Langages & systèmes

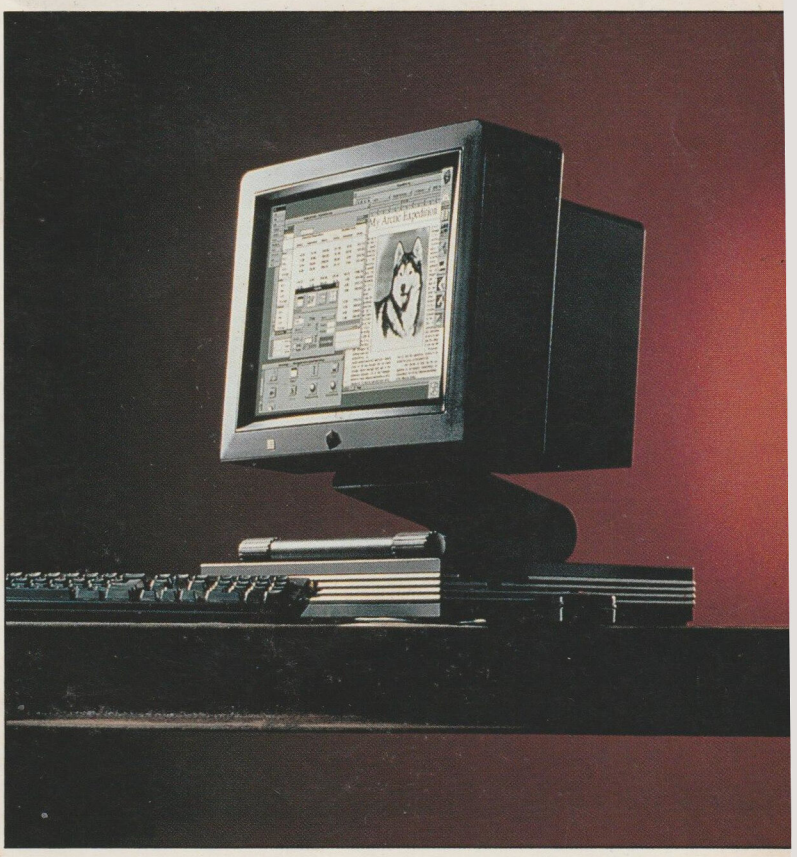
LE MAGAZINE DE LA MICRO-INFORMATIQUE PROFESSIONNELLE
JUILLET-AOÛT 1992 - N° 8 SUPPLÉMENT DÉVELOPPEURS DU N° 83 D'INFOPC ISSN 0951- 6402

GOLD 92

LE GRAND OPEN LOGICIEL ET DÉVELOPPEMENT

SPÉCIAL NEXT

- Station NeXT : la plate-forme de développement idéale
- NeXTStep 3.0 : programmer sans douleur

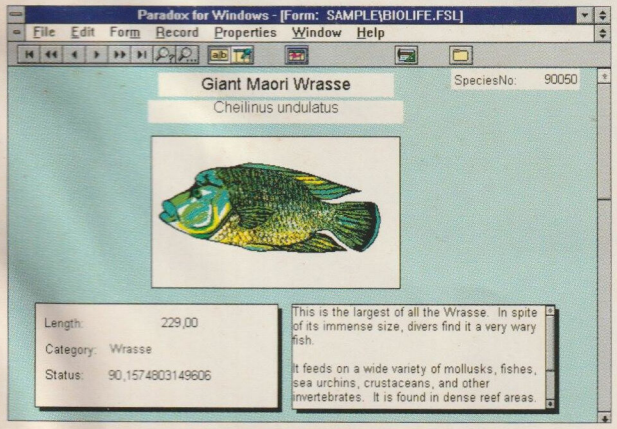


NANTUCKET TOOLS

- Dopez votre Clipper

OBJECT PAL

- Paradox se met à l'objet



MSJ WINDOWS 3.1

Performance au rendez-vous

ET AUSSI...

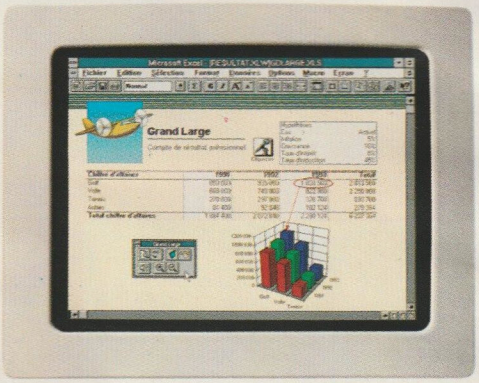
- LE C++ SANS OBJET
- LES INDEX DE FICHIERS
- OBJECT WINDOWS
- LES PROTOCOLES DE COMMUNICATION

NE PEUT ÊTRE VENDU SÉPARÉMENT

Nouveau Microsoft Ex
il anticipe, vous

Simplifiez vos travaux quotidiens ! Excel, le tableur le plus vendu en France, inaugure avec la version 4.0 une nouvelle génération de tableurs : pour vous faciliter la tâche, il prend des initiatives ! Il anticipe vos besoins, déduit les informations de façon logique, propose une assistance au moment opportun... A croire qu'il devine vos pensées ! Des tâches qui auraient pris des heures sont réalisées en un temps record !

Tableur minute. Une minute suffit pour réaliser un tableau et le mettre en page. Frappez "janvier" et étirez la sélection avec la souris : Excel indique "février, mars, avril..." Si vous placez les chiffres 100 et 200 dans deux cellules, la même action l'amène à déduire les valeurs suivantes : 300, 400, 500... Un clic sur l'icône Somme automatique fait apparaître les totaux. La mise en page est tout aussi aisée avec la fonction Format automatique qui dote votre tableau de couleurs, motifs et encadrements stylés.



Suivez le guide. Les Assistants vous guident pas à pas dans la création de tâches complexes - une excellente façon d'apprendre en travaillant. Les barres d'outils donnent accès aux fonctions les plus couramment utilisées. Selon le travail envisagé, affichez celles qui semblent appropriées : calcul, graphiques, dessin... ou bien créez vos propres barres.

Analysez, synthétisez, communiquez. Excel 4.0 est l'outil idéal du décideur.
Le Gestionnaire de scénarios permet de simuler plusieurs hypothèses, et de produire un état synthétisant les résultats obtenus.
L'Assistant Tableaux croisés produit un tableau résumé à partir des valeurs d'une base de données.

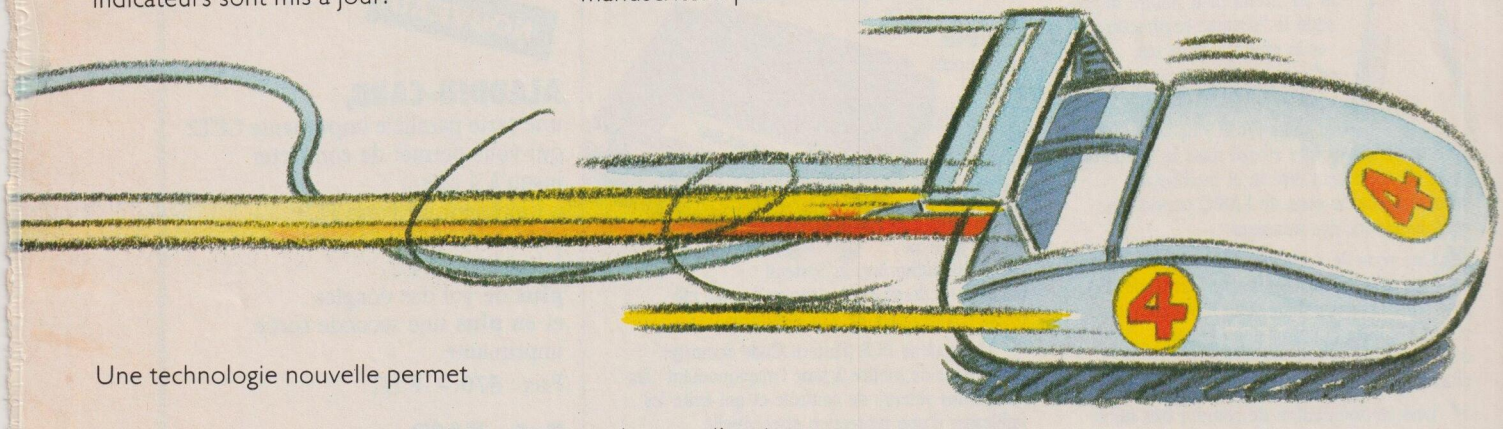
cel 4 pour Windows : allez plus vite.

Le Solveur résoud les problèmes d'optimisation les plus complexes. Les Classeurs rassemblent vos documents par thème dans un seul fichier, et facilitent les consolidations. Celles-ci sont interactives : si une information change, tous vos indicateurs sont mis à jour.

graphiques pour mettre en valeur vos analyses.

Le tableur qui sait lire et parler. Excel 4.0 est ouvert aux nouvelles extensions Pen et Multimedia de Windows. Il peut donc recevoir ses données sous forme manuscrite. Il permet

Profitez jusqu'au 31 août de la mise à jour à 990 F TTC au lieu de 1190 F TTC. Prix public conseillé 4990 F HT.



Une technologie nouvelle permet à Excel d'exécuter directement les macros Lotus® 1-2-3.®* Avec le Diaporama, créez des "shows" avec effets spéciaux à partir de vos données et enrichissez vos présentations de superbes

également d'agrémenter vos informations de sons et commentaires vocaux. Mais quel est donc le secret de ce tableur surdoué ? Simplement le fait qu'il est né de vos innombrables suggestions !

Pour toute information complémentaire :
3616 Microsoft
pour consulter
la liste des revendeurs agréés.
Le Service Clients : 16 (1) 69 29 11 11
Microsoft : Infomart CNIT - La Défense
MICROSOFT - 91957 LES ULIS CEDEX

Microsoft®

LES LOGICIELS QUI DONNENT DES AILES.

* Compatibles Lotus version 2.01.

Toutes les marques citées sont des marques déposées.

Service lecteurs n° 1 (dernière page)

★ ROUX SEGUELA CAYZAC & GOUDARD

AVANT DE CHOISIR UNE PROTECTION DE VOS LOGICIELS...

...contre le piratage et l'utilisation frauduleuse, vérifiez que le système de protection envisagé possède bien les qualités suivantes :

UNE CLÉ AU HARDWARE EXCEPTIONNEL

Les protections hardware sont maintenant devenues un standard incontournable dans le monde entier. Cependant toutes les clés ne sont pas égales. Une clé se doit de posséder les qualités suivantes :

✔ Compatibilité et transparence. La clé doit pouvoir fonctionner sans problème sur tous les ordinateurs de vos clients. L'utilisateur devrait pouvoir l'oublier dès sa connection.

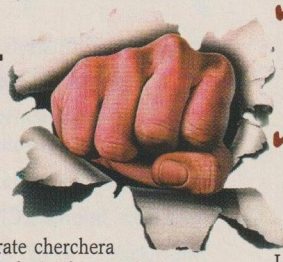
✔ Une électronique sans faille. Un circuit ASIC (spécifiquement conçu et fabriqué) devrait être inclu dans le hardware. Seul un tel circuit peut rendre impossible le "reverse engineering" et le piratage de la clé.

✔ Un code développeur unique et inaccessible doit être gravé dans l'ASIC. Le code client ne devrait jamais être stocké dans la mémoire de la clé où il peut être lu et modifié à volonté... Seul le code de l'ASIC permet une reconnaissance des messages.

✔ Une zone de mémoire disponible en lecture/écriture doit être disponible sur demande. On doit pouvoir écrire dans cette mémoire depuis n'importe quel PC, sans avoir besoin d'équipement spécial (sérialisation chez le client).

✔ Une consommation de courant très faible permet à la clé de fonctionner dans les pires conditions, du PC au laptop, avec ou sans imprimante connectée, allumée ou non...

UN LOGIGIEL PERFORMANT



Comme il est pratiquement impossible de dupliquer ou de pirater une clé qui possèderait toutes les qualités citées, le pirate cherchera la faille au niveau du logiciel protégé. Et afin de prévenir cela, le logiciel de protection se doit de posséder aussi les qualités suivantes :

✔ Un Module de Protection à Linker (MPL) intégré à votre programme source, afin de pouvoir insérer des appels à la clé depuis n'importe quel endroit du programme protégé.

✔ Un système "d'enveloppe" autour de votre programme exécutable. Votre programme protégé peut être verrouillé par cryptage et ne sera décrypté puis exécuté que si la présence de la clé est vérifiée...

✔ Des mécanismes sophistiqués de protection, des pièges antidébugging, des cryptages/décryptages dynamiques, du code auto-généré et auto-modifiable sont des fonctionnalités importantes qui augmentent la sécurité de votre protection.

HASP, LE SYSTÈME PROFESSIONNEL D'ALADDIN Knowledge Systems VOUS OFFRE TOUTES CES FONCTIONNALITÉS ET PLUS ENCORE :

Hasp a été conçu par un groupe d'experts informatiques, cryptologues professionnels et ingénieurs en électronique. Il en résulte que les clés HASP sont supportées par ce qui est probablement le meilleur software disponible sur le marché. De plus, le système HASP a été testé avec succès sur une vaste gamme d'ordinateurs. De surcroît, HASP vous offre :

✔ Le système FAS (Full Authorization System) qui permet de protéger avec une seule clé MemoHASP-4 près de 112 progiciels.

✔ Le système PCS (Pattern Code Security) permettant de mettre à jour "magiquement" les différentes valeurs de contrôle et qui évite les faiblesses d'une protection séquentielle.

✔ Un système de détection de virus qui peut être incorporé dans le programme protégé.

✔ Les HASP sont compatibles avec un grand nombre de systèmes d'exploitation comme DOS, SCO Xenix, et Unix-386, OS/2, AIX, PHAR-LAP, ERGO et RATIONAL DOS Extenders et aussi avec WINDOWS et AUTOCAD.

✔ NetHASP, la protection ultime : protégez les licences de dizaines de logiciels DOS ou Windows 3.0 fonctionnant en réseau avec une seule clé. Les réseaux supportés sont NOVELL (dédié ou non), Lan Manager, Lantastic, Banyan, DLink et tous les logiciels basés sur Net BIOS

LORS D'UN TEST EXÉCUTÉ DANS UN LABORATOIRE INDÉPENDANT ALLEMAND, LE HASP ETAIT LA SEULE CLÉ A NE PAS ETRE DÉPLOMBÉE.

(CT Magazine, RFA, Mai 1990)

ET FINALEMENT :

Il est indiscutable que les HASP vous offrent le meilleur rapport qualité/prix du marché.

Dans les sept dernières années, HASP fabriqué par

ALADDIN KNOWLEDGE SYSTEMS LTD.

a permis à des milliers de développeurs et entreprises, parmi lesquelles plusieurs "Fortune 500", de protéger leurs logiciels efficacement.

NOUVEAU

ALADDIN-CARD,
une carte parallèle imprimante CPT2 qui vous permet de connecter jusqu'à 4 clés
à l'intérieur de votre PC :
plus de longues chaînes
à l'arrière du PC,
plus de vol des dongles
et en **plus** une seconde sortie imprimante.
Prix : **570.–** F HT

Data HASP V3.0
pour la protection des banques de données est maintenant disponible :
Prix : **950.–** F HT
pour une licence sur site illimitée.

Nouvelles séries de
NetHASP V3.0
(protection des logiciels en réseau).
NetHASP-5, 10, 50, 100 et plus.

Très importants rabais pour des commandes de 500 et +

DISTRIBUTEUR POUR LA FRANCE

LOGIDATA SA

B.P. n° 2 - Publier - 74500 EVIAN / France
Tél. 50 70 73 75 - Fax 50 75 31 44

DEVELOPPEZ + VITE VOTRE INTERFACE ECRAN

HIGH SCREEN 5.5 pour DOS, OS/2, UNIX et WINDOWS

Développez 10 fois + vite !

Fonctionne dans **tous** les langages; PAS de résidents!

HIGH SCREEN 5.5 est-il le meilleur générateur d'interfaces écran du monde?

Il est difficile de répondre à cette question en une seule page! Les fonctionnalités complètes de HIGH SCREEN 5.5 occupent à elles seules 8 pages! Le fait que plus de 10.000 informaticiens aient choisis HIGH SCREEN en France et que plusieurs centaines de milliers d'applications développées avec HIGH SCREEN soient en service apporte déjà un premier élément de réponse. Les illustrations ci-contre donnent un autre élément de réponse: **Tout ce que vous voyez ci-contre est réalisable dans votre langage, très facilement et très vite.**

Voici d'autres éléments de réponse:

- Tout d'abord HIGH SCREEN fonctionne sous DOS, WINDOWS, OS/2 et sous UNIX, *avec un code source compatible!*
- Ensuite, la quasi totalité des langages du marché est supportée: C, C++, Pascal, Basic, Nantucket, dBase, FoxPro, Fortran, Cobol, ADA, Lisp, Dataflex, Paradox Engine, etc...
- Le principe de développement avec High Screen 5.5 est simple et efficace:

- étape 1: sous un éditeur WYSIWYG, vous créez vos objets (masque, fenêtre, champ, icone, menu déroulant, prototype,...); tout ce qui est fait sous l'éditeur ne sera plus à faire en programmation! Vous définissez également lors de cette étape les tests à effectuer automatiquement sur les champs .

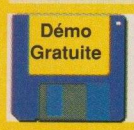
- étape 2: directement sous l'éditeur, vous testez le résultat que vous obtiendrez dans votre programme.

- étape 3: vous programmez avec les objets créés,dans votre langage,à l'aide d'ordres simples.

- étape 4: vous diffusez librement les superbes applications que vous avez réalisées!

- La programmation est moderne, puissante, en français, facile à comprendre et à maintenir. Par exemple, la seule ligne de code ECRAN,SAISIE effectue les traitements suivants: active la souris, active la saisie avec passage automatique d'un champ à l'autre, effectue les tests sur les champs, gère l'aide contextuelle; et retourne au programme des valeurs pré-testées !
- En plus du générateur d'écrans, HIGH SCREEN propose en standard: un générateur de prototype sans programmation, un générateur de code source, un éditeur d'icones, un générateur de menus déroulants, un éditeur de dossier, un mini-grapheur, un outil de capture d'écrans...
- Les Utilisateurs de HIGH SCREEN nous disent **diviser par 10** leurs durées de **développement.**

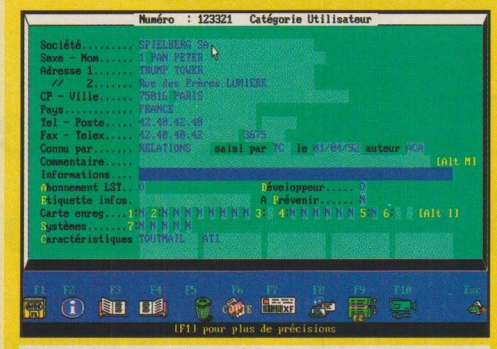
Demandez votre disquette de démo: vous devez connaître HIGH SCREEN 5.5 pour être plus rapide!



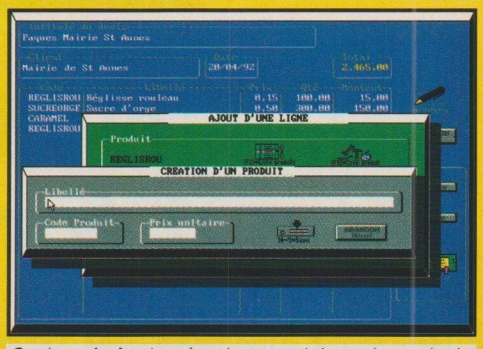
Appelez de suite pour recevoir votre disquette d'évaluation gratuite !
Vous pouvez également faxer votre demande, ou utiliser le minitel.

Une superbe application en mode graphique vraiment facile à réaliser, ici sous **DOS**

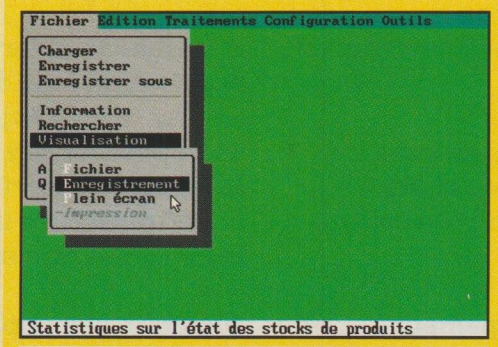
La *même* application ayant simplement été *recompilée* sous **Windows**: *incroyable, votre code source est compatible!*



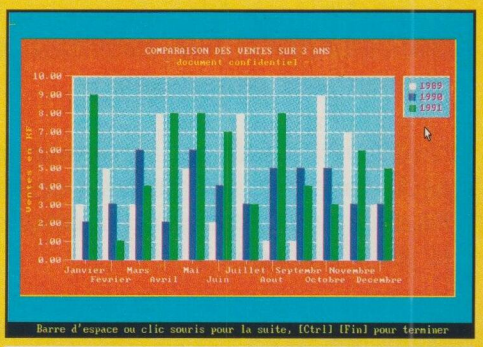
Voici un écran de saisie contrôlée, avec de nombreux champs et des icones, sous DOS.



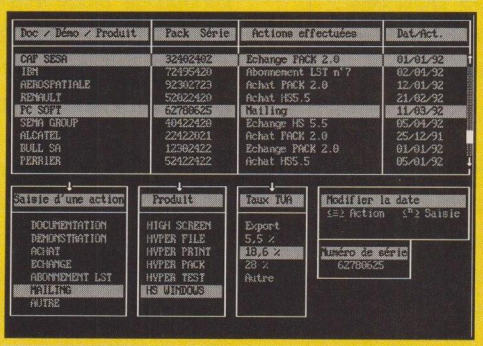
Ouvrir une fenêtre (en gérant le contexte) demande une simple ligne de code ("OUVRE") quel que soit votre langage!



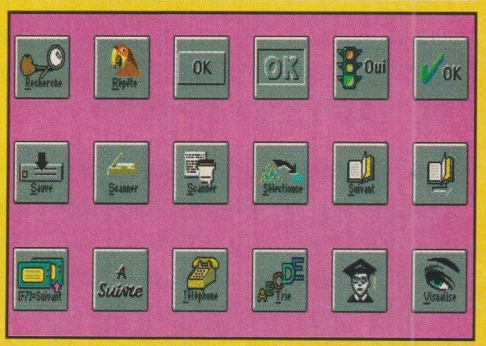
Les menus déroulants sont réalisés très vite (quatre minutes pour celui-ci); le "fond" peut être une image, un écran ou rien,comme ici.



Un mini-grapheur (histogrammes, courbes et camemberts) permet d'afficher des graphiques en une seule ligne de code !



HIGH SCREEN permet également de développer de superbes applications pour le mode "texte monochrome".



Même sous **DOS**, vos programmes disposeront des icones, sans modifier votre programmation habituelle...

Service lecteurs n° 4 (dernière page)

Marques déposées par leurs propriétaires respectifs. Document non contractuel. Prix modifiables sans préavis.

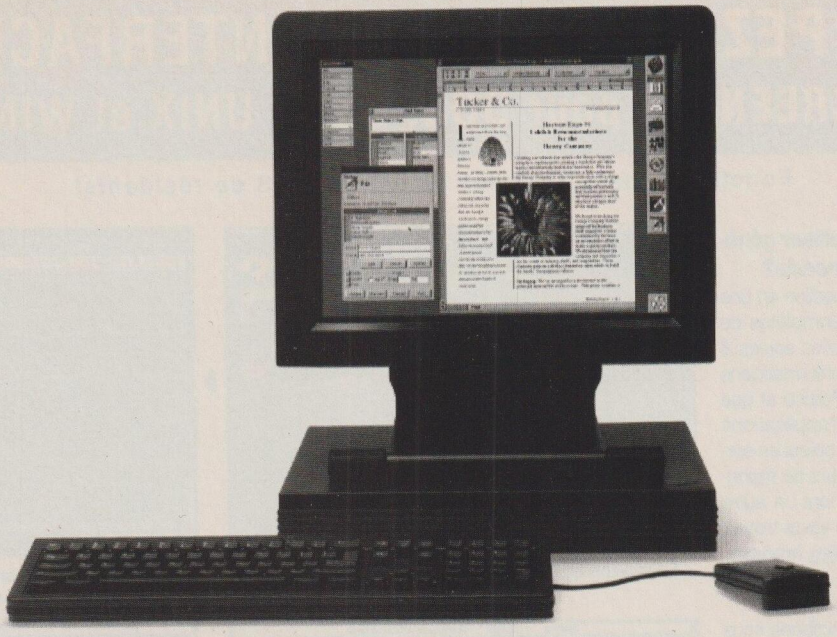
PC SOFT

Siège Montpellier : 216 rue des Escarceliers, BP 3019 34034 MONTPELLIER
Tél: **(16) 67.032.032** Fax (16) 67.03.07.87 Technique: (16) 67.03.17.17

Agence Paris: 34 Bd Haussmann 75009 PARIS
Tél: **(1) 48.01.48.88** **Minitel** 3614 code PC SOFT

n° 1 en France

Service lecteurs nº 5 (dernière page)



NeXT pense à ceux qui écrivent des logiciels

Grâce aux interfaces graphiques, les logiciels n'ont jamais été aussi faciles à utiliser. En revanche, leur développement, leur maintenance et leur évolution sont devenus de plus en plus complexes.

NeXT est le seul constructeur à proposer une plate-forme complètement orientée objet, fondement d'une programmation plus rapide et plus efficace.

Notre environnement de développement NeXTSTEP est véritablement orienté objet jusqu'aux couches systèmes. Vous pouvez ainsi écrire des programmes élaborés et intuitifs jusqu'à dix fois plus rapidement que sur d'autres plates-formes.

L'Application Kit™ de NeXTSTEP fournit l'interface et les objets systèmes communs à la plupart des applications utilisables directement et sans modifications. En plus des objets disponibles que vous pouvez modifier, il vous est possible d'en créer de nouveaux aussi simples à utiliser.

Interface Builder™ facilite la conception des applications qui fonctionneront exactement comme vous l'aviez rêvé. A l'aide de la souris vous positionnez et redimensionnez vos éléments d'interface, changez leurs attributs et définissez les interactions entre les objets, sans écrire une seule ligne de code.

Bien sûr, les applications développées avec NeXTSTEP présentent de multiples avantages propres aux stations de travail NeXT. Elles offrent un graphisme exceptionnel, sont faciles à utiliser et cohérentes, donc ne nécessitent qu'une formation très courte.

En plus d'une parfaite intégration à votre environnement de travail, vos applications communiqueront avec le courrier électronique multimédia et le nec plus ultra en matière de logiciels disponibles.

Si vous souhaitez en savoir plus et recevoir une documentation complète, renvoyez ce coupon à :
NeXT Computer France, 175 rue Jean-Jacques Rousseau, 92138 Issy Les Moulineaux Cedex

Société : _____ Fonction : _____

Nom : _____ Prénom : _____

Adresse : _____

Code Postal : _____ Ville : _____

NeXT®

Sommaire

- 9 -

Re-sources

Une sélection des nouveaux outils de développement.

- 16 -

La station NeXT

La plate-forme de développement idéale existe, nous l'avons essayée.

- 28 -

ObjectPAL

Paradox Windows intègre un langage orienté objet qui n'a plus grand-chose de commun avec PAL sous DOS.

- 38 -

Sur les traces de NeXTStep 3.0

NeXTStep renforce dans sa version 3.0 sa réputation de meilleure interface de développement.

- 52 -

Object Windows

Cette bibliothèque d'objets accompagne le Borland C++.

- 58 -

Windows 3.1

La dernière mouture de l'interface de Microsoft apporte quantité d'améliorations.

- 84 -

Le C++ sans objet

Le C++ est avant tout un meilleur C.

- 88 -

Nantucket Tools II

Cette bibliothèque de routines ouvre Clipper sur des applications pour lesquelles il n'était pas conçu au départ.

- 96 -

Turbo Vision

Turbo Vision dans la pratique.

- 100 -

Les index de fichiers à base de B-arbres

Les algorithmes d'index à base de B-arbres se distinguent par leurs performances.

- 112 -

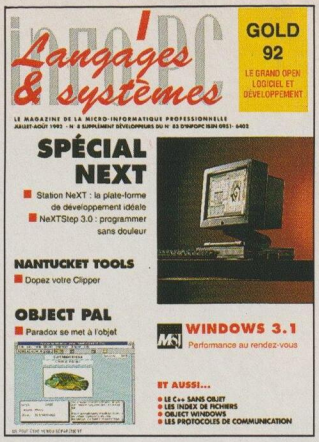
Les protocoles de communication

Une bonne connaissance des protocoles devient aujourd'hui indispensable pour ouvrir une application sur l'extérieur.

- 120 -

Concours GOLD

Dernier rappel. Ne laissez pas passer votre chance de participer au Grand Open Logiciel et Développement 92.



NeXT, l'outsider

Microsoft commence à diffuser des versions pré-bêta de Windows NT tandis qu'IBM dépense une fortune en publicité pour convaincre l'utilisateur de franchir le pas pour OS/2 2.0. L'un en est encore dans sa phase de mise au point et l'autre trône sur les étalages depuis quelques mois à peine. Quant au principal intéressé, le développeur, il se retrouve dans une situation similaire à celle de 1982, date de la commercialisation du PC : il peut compter sur les doigts d'une main les langages disponibles pour ces systèmes.

Borland devrait annoncer, dans le courant de l'été, C++ et Object Vision pour OS/2 2.0, mais le disert Philippe Kahn reste très discret sur le portage de Turbo Pascal, plus populaire que le C du même éditeur. Quant à Paradox et à dBase, Borland «attend de voir». En clair, sa décision dépendra des parts de marché gagnées par OS/2 2.0 sur MS-DOS et sur Windows 3.10. Microsoft, pour sa part, ne se précipite pas pour effectuer le lifting nécessaire à ses compilateurs existants pour OS/2 version 1.x. Bill Gates s'efforce plutôt de combler son retard face à Borland en acquérant FoxPro et en commercialisant le C++/C 7.0.

En regard, il existe une plate-forme dans laquelle le développeur dispose de tous les outils pour construire des applications en un temps record : la station NeXT. Celle-ci a vu le jour en octobre 1988 et a été présentée par Steve Jobs, son fondateur, comme le «*meilleur ordinateur de tous les temps*». Elle a été rattrapée depuis sur le plan technologique mais conserve une longueur d'avance avec Step, l'interface graphique qui habille son système d'exploitation multitâche va bientôt pouvoir s'exécuter sur les PC/486 et devenir, du même coup, une solution de remplacement de Windows NT et d'OS/2 2.0, avec des outils de développement parfaitement rodés et un système multitâche qui a fait ses preuves : Unix.

Yves Guittard

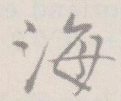
C-scape : un véritable plaisir.

Gestionnaire d'interfaces écran graphique et texte

Service lecteurs n° 6 (dernière page)

Le système de gestion d'écrans C-scape® est une bibliothèque de fonctions en langage C permettant la saisie et la validation de données à l'écran, la gestion de menus et fenêtres, l'édition de texte, et les systèmes d'aide en ligne. L'outil de conception d'écrans de C-Scape, Look & Feel™, vous permet de créer des écrans de façon interactive et de générer le code source correspondant.

C-Scape comprend des fonctions de haut niveau facilement modifiables, aussi bien que des primitives permettant de construire de nouvelles fonctions. Grâce à l'architecture orientée objet de C-Scape, vous développerez des applications plus fonctionnelles, plus flexibles, plus portables, et vous aurez plus de plaisir à le faire.



Un produit phare dans l'industrie du logiciel. Depuis 5 ans, plusieurs milliers de développeurs dans le monde entier se sont convertis à C-Scape, abandonnant leur bibliothèque "maison", ou des produits peu flexibles. La presse approuve: "C-Scape est de loin le meilleur. ...un véritable plaisir", ecrit IEEE Computer. Les plus grandes sociétés ont selectionné C-Scape comme standard de développement.

C-Scape peut être utilisé avec des SGBD, des bibliothèques graphiques, ou toute bibliothèque C ou C++. De plus, C-Scape peut combiner texte et graphique. Pour porter votre application de DOS ou OS/2 vers UNIX, VMS, AIX, recompilez simplement. C-Scape supporte également les "DOS extenders" Phar Lap, Rational et Ergo, ainsi que MetaWINDOW.

Graphique. Fonctionne en couleur, en mode texte ou graphique. Peut lire des images graphiques depuis des fichiers PCX.

Architecture orientée objet. Ajoutez vos propres fonctions et créez des modules de code réutilisables.

Support de la souris complètement intégré pour la selection des options de menu, les champs de saisie, et pour déplacer et redimensionner une fenêtre.

Portabilité. Le code de C-Scape est indépendant du hardware. Supporte DOS, OS/2, UNIX, AIX, VMS, QNX, XENIX... Détecte automatiquement Hercules, CGA, EGA, VGA, et les modes 43 et 50 lignes.

Edition de texte. Comprend un éditeur de texte avec contrôle de la césure, fonctions rechercher/remplacer, et manipulation des blocs de texte.

Flexibilité des champs. Champs masqués, protegés, marqués, avec label, obligatoires, avec ou sans echo. Validation complète des données. Champs de type heure, date, montant, liste déroulante, et autres fonctions de haut niveau. Créez votre type de champ.

Fenêtres. Encadrées, pop-up, superposées, eclatées, ascenceur, défilement horizontal; taille et nombre des fenêtres limités par la memoire RAM uniquement.

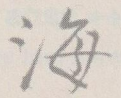
Menus. Déroulants, hiérarchiques, pop-up, style Lotus 1-2-3, ou créez le vôtre.

Système d'aide contextuel. Prompts et messages d'aide attachés à un écran ou à un champ. Messages à références croisées pour créer un système d'aide en ligne de style hypertexte.

Gestion des écrans. Ecrans stockés dans des fichiers ou linkés avec l'application.

Génération de code. Créez n'importe quel écran avec l'outil interactif Look & Feel, testez-le, puis convertissez-le automatiquement en code C.

Essayez C-Scape ! C-Scape est *puissant, flexible, portable*, d'apprentissage et d'utilisation facile. C-Scape est livré en standard avec la totalité du code source de la bibliothèque et l'outil de création d'écrans Look & Feel. La documentation est claire, extensive et detaillée. Avec C-scape, pas de module résident et pas de royalties sur les applications diffusées.



C-Scape est un produit d'**Oakland Group**. Smartline assure le support technique de C-scape par téléphone et par fax ainsi que des sessions de formation sur le produit.

Appelez-nous au (1) 46 34 64 19
ou envoyez le bon de commande ci-dessous par fax ou par courrier. Après le plaisir de C-Scape, le développement ne sera jamais plus ce qu'il était.

C-scape et Look & Feel **version 3.2** : DOS, OS/2 (Borland, Microsoft, Zortech): 4500FHT; UNIX: 13500FHT; Autres OS: nous consulter. Cartes de crédit acceptées.

smartline

30, rue Saint André des Arts
75006 Paris France
Fax: (1) 40 46 85 89 ***Tél: (1) 46 34 64 19***

Oakland Group, Inc 675 Massachusetts Ave, Cambridge, MA 02139 USA.Tél: 617-491-7311. Les produits d'Oakland sont distribués en Australie, Belgique, Danemark, Autriche, Allemagne, Suisse, Norvège, Suède, Angleterre, France. La photographie représente un programme C-Scape combinant une saisie de données avec des images vidéo chargées depuis un fichier .PCX. C-Scape et Look & Feel sont des marques déposées par Oakland Group, Inc.; les autres marques citées appartiennent à leurs priopriétaires respectifs. Les fonctionnalités, prix et conditions sont sujets à modification. Documentation en anglais. C-scape (seascape): paysage de mer.

✂

❑ Je désire recevoir une documentation sur C-scape

❑ Je désire recevoir le dossier d'évaluation comprenant les deux disquettes de démonstration C-scape et Look & Feel. Ci-joint un chèque de 120 F TTC. Une facture sera jointe à l'envoi.

nom prénom _____ société _____ tél _____

adresse _____

Re-sources

File System

Buro service annonce la sortie de File System, système de gestion de fichiers séquentiel indexé, accessible aux développeurs Windows.

File System autorise le traitement de fichiers de grande taille (un milliard de fiches) et, grâce à son utilisation d'index organisés en B-Arbres, offre une bonne vitesse d'accès aux données. Il est accessible au travers du langage de développement et se présente comme de simples appels à des fonctions. De plus, File System importe les fichiers au format dBase et les convertit au format CSV (Word pour Windows) ou ASCII tabulé pour Excel. Son prix est de 990 F HT. Edité et distribué par Buro service.

File System

• Editeur : Microprocess • Distributeur : Microprocess • Réseau industriel pour la connexion de capteurs et de calculateurs industriels • Gestion d'une base de données temps réel répartie • Hauts débits de transmission • 24 000 objets identifiables •

SoftWorks

Conçu et distribué par Microdata Soft, SoftWorks est un atelier de développement logiciel croisé C et objets C++, orienté pour les applications temps réel. Il couvre les principales phases du cycle de vie normalisé, de la conception (concepts objets) au codage puis aux tests unitaires et d'intégration.

L'environnement unifié de SoftWorks est identique sur un grand nombre de configurations : stations Unix Sun, Delta, HP, PC/386 (Xenix, Venix...) ou calculateurs industriels sous OS-9.Tous les outils de SoftWorks peuvent être utilisés, soit avec un terminal alphanumérique, soit à travers une interface graphique exploitant X Window 11.4. Il permet aussi de produire du code 68000 à 68040 et des microcontôleurs CPU32.

SoftWorks comprend également un débogueur symbolique travaillant aussi bien en C qu'en C++. Enfin, SoftWorks est indépendant du noyau temps réel cible. Il supporte l'écriture de programmes tournant sous OS9, VRTX32, VMEexec... ou sur une machine nue (sans exécutif) avec SoftKernel, le noyau temps réel intégré.

FIP

Développé et distribué par Microprocess, le réseau FIP (Flux information processus) est prévu pour permettre l'inter connexion de capteurs et d'actionneurs avec des calculateurs (VME ou PC, automates programmables, régulateurs...). Son bus est ouvert : toute station peut être productrice de toute variable, et consommatrice de n'importe quelle autre. FIP permet une gestion du processus avec une base de données temps réel répartie (température d'une cuve, état d'une vanne...). La longueur maximale du réseau est de 2 km, et 256 stations au maximum peuvent y être connectées. Le support physique est de type paire torsadée ou fibre optique en code Manchester II. Les débits sont de 31,25 kbits/s, de 1 ou 2,5 Mbits/s. Au niveau de la couche liaison de données, 24 000 objets sont identifiables, et FIP prévoit des échanges cycliques de variables par tables configurables et des accès sur demande à toutes les variables.

La transmission de messages se fait selon le protocole LCC type 3 d'IEEE 802.2 et garantit des instants de mise à jour et des temps de transfert rapides. Pour finir, quelques chiffres relatifs aux performances de ce réseau : 128 valeurs analogiques, produites par 128 capteurs, sont transmises en 22 ms, tandis que 512 états TOR, diffusés par 16 stations comportant 32 entrées chacune, sont émis en 2,5 ms.

FIP

• Editeur : Microprocess • Distributeur : Microprocess • Environnement de développement C ou objet avec C++ • Environnement de développement unifié multi plateforme • Indépendant du noyau temps réel cible • Débogueur symbolique pour sources C ou C++ •

Databoss 3.5 B

Innosoft propose aujourd'hui un nouvel outil de développement nommé «Databoss version 3.5 B». Ce générateur complet a été conçu pour le développement d'applications en pascal ou en C/C++.

Il permet de produire tous les éléments d'une application comportant toutes les caractéristiques professionnelles et de créer rapidement des versions de démonstration des applications en limitant le nombre d'enregistrements dans les fichiers. De plus, les programmes engendrés peuvent fonctionner en mode mono et multi-utilisateur.

Pour réaliser une application en réseau, il suffit de sélectionner l'option adéquate. Databoss se charge du reste (le verrouillage et la protection des enregistrements). Ce programme génère du code source pascal ou C.

Il possède un générateur de rapports Wysiwyg et est associé à un compilateur version 3.5 B qui permet d'obtenir des fichiers .EXE compilés. Il est livré avec un exemple complet (système de gestion) développé en Turbo Pascal. Pour finir, précisons que les applications écrites ne sont soumises à aucunes royalties. Databoss en version française coûte 7 995 F HT.

Databoss 3.5 B

• Editeur : Innosoft • Distributeur : Innosoft • Générateur d'applications en pascal, en C ou en C++ • Applications fonctionnant en mode mono ou multi-utilisateur • Générateur de rapports Wysiwyg • Pas de royalties •

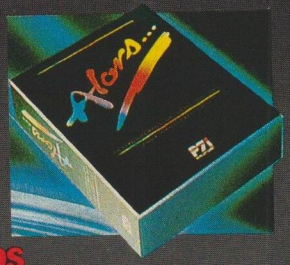
Kits de 16 et 64 Mo pour Quadra

MSG propose des kits de 4, 16 ou 64 Mo pour la gamme Quadra d'Apple. Composés de barrettes de 16 Mo, ils s'insèrent directement sur la carte mère. La mémoire d'un Quadra peut ainsi être accrue à 128 Mo. Les prix publics vont de 2 490 à 34 560 F HT selon la capacité.

Alors... MD

UN AGL POUR NANTUCKET 5

- Dictionnaire de données
- Générateurs de sources
- Bibliothèque



La maîtrise du temps

Alors... intègre trois modules complémentaires : un dictionnaire de données, des générateurs de sources et une bibliothèque, destinés à aider le développeur à gagner du temps et créer des sources très lisibles.
A l'image des moteurs relationnels, le dictionnaire de données permet de créer les fichiers, les index, les contrôles, les liens interfichiers et d'en garantir la cohérence hors des lignes de programmes.

Un très petit nombre de lignes à maintenir

Le dictionnaire élimine toute l'écriture des contrôles interfichiers. La bibliothèque comporte une cinquantaine de fonctions majeures pour les développeurs (attribution d'un nom de fichier temporaire en réseau, log d'erreur automatique, recherche d'un champ dans un autre fichier...). Les générateurs développent un ensemble de fonctions modulaires (entrées/sorties, menus déroulants, doc. technique...).

Une maintenance aisée

Voici quelques exemples :
Un champ change de longueur? Tous les éléments utilisant ce champ fichiers, index, écrans et variables mémoires sont modifiés en cascade et reconstitués dynamiquement.
Un index est nécessaire pour un nouvel état? Vous le déclarez dans le dictionnaire et tous les programmes déjà écrits le reconnaissent immédiatement et le prennent en compte.
Pour en savoir plus renvoyez-nous le coupon ci-dessous.

PZI DEVELOPPEMENT
88 avenue Kléber - 75116 Paris
Tél. 47.27.01.29 ■ Fax 47.04.66.08

OFFRE SPECIALE 8 900 F HT 23 140 F HT 42 900 F HT
Version Monoposte ~~15 000 F HT~~ ■ 5 p. ~~39 000 F HT~~ ■ 10 p. ~~78 000 F HT~~

Je souhaiterais ❑ Une documentation.
❑ Le manuel de découverte avec la disquette de démonstration. Ci-joint un chèque de 177,90 F TTC.
❑ Assister à une démonstration gratuite.

Nom _____ Fonction _____
Société _____ Tél. _____
Adresse _____

Nantucket est une marque déposée par Nantucket Corporation

Service lecteurs n° 7 (dernière page)

Re-sources

Codesafe 3.19

Le CTI propose aujourd'hui la version 3.19 de Codesafe, un système de protection par disquette contre la duplication et l'exécution illégales de logiciels.

Ce système de protection par disquette constitue une solution de remplacement intéressante au système de protection par clé. Le coût de l'opération est 4 à 5 fois moins élevé, et la protection par disquette permet de faire cohabiter plusieurs programmes préservés sur un même disque dur. Celle-ci est réalisée grâce à un code clé se trouvant sur la disquette ou sur le disque après installation. Chaque disquette est marquée d'un code clé unique de huit caractères et de trois signatures électroniques non reproductibles.

Codesafe 3.19

• Editeur : Eliashim. • Distributeur : CTI • Système de protection par disquette • Clé de huit caractères • Trois signatures électroniques non reproductibles • Immunité contre les virus • Transparence pour l'utilisateur •

Cette marque est réalisée par un appareil non-DOS, ce qui permet de la rendre indupliquable par un lecteur DOS. Grâce à ce dispositif, les déplombeurs «Ghost» comme Unguard se retrouveront impuissants de même que tous les utilitaires de copie de la création.

Il suffira de lancer Codesafe avec les options de configuration désirées (cryptage, antivirus, multiples installations...) et d'effectuer la protection, l'utilisateur n'intervenant pas dans le fonctionnement de cette dernière. Exemple de prix : 40 F HT par disquette pour 1 000 disquettes par an.

Activator / ws

Micro Sigma, distributeur de la société Software Security en France, annonce la disponibilité de la nouvelle clé de protection de logiciels pour station de travail : Activator / ws.

Elle permet au développeur d'effectuer une protection efficace des conditions d'exécution de logiciels tout en donnant à l'utilisateur la liberté du système sur lequel l'application tourne.

Cette technique offre une solution efficace du contrôle d'exécution de programmes sur réseau. Quatre clés composent la gamme, une clé simple de contrôle d'exécution, une autre qui comprend une mémoire non volatile programmable depuis l'application, une troisième à mémoire spécialisée réseaux et, enfin, une clé de capsulation qui permet de contrôler l'exécution du programme dont l'utilisateur n'a pas les sources.

Cette dernière se révèle intéressante dans les services contenant des parcs micros importants afin d'empêcher la prolifération de copies de programmes.

Toutes ces clés sont disponibles chez Micro Sigma pour MacOS, DOS, OS/2, Xenix, Unix 386 et Sun.

Activator / ws

• Editeur : Software Security • Distributeur : Micro Sigma • Protection de logiciels • Contrôle d'exécution des programmes sur réseaux • Disponible pour MacOS, DOS, OS/2, Xenix, Unix 386 et Sun •

Re-sources

ObjectCenter

Centerline Software annonce la disponibilité d'ObjectCenter 1.1 pour les stations de travail de Digital Equipment Corporation, l'environnement de programmation Unix en langage C++ déjà disponible sous Sun. Distribué en France par Graphael, ObjectCenter est un environnement de programmation complet regroupant tous les outils nécessaires à la création, au déboguage, aux tests et aux enrichissements d'applications.

Notons sa compatibilité avec la plupart des bases de données orientées objet et le support du Softbench de HP. Adapté aux projets de développement complexes, ObjectCenter détecte plus de 250 erreurs au cours du chargement et de l'exécution d'un programme, et son système de déboguage permet d'en identifier les origines.

De plus, ObjectCenter intègre des "Browsers" graphiques pour faciliter la compréhension des applications. Les utilisateurs peuvent tester et évaluer des fonctions tout en les codant et se servir des outils dont ils disposent déjà sans quitter l'environnement d'ObjectCenter.

ObjectCenter

• Editeur : Centerline Software • Distributeur : Graphael • Environnement de programmation en C++ pour stations DEC • Disponible sous Sun • Compatible avec les SGBD orientés objet • Détection des erreurs • Système de débogage • «Browsers» graphiques • Espace interactif de travail •

Lotus supporte Twain

Lotus Development vient d'annoncer le support du standard d'acquisition d'images Twain qui sera inclus dans les fonctions de traitement de Lotus Notes, logiciel de communication de groupe.

Le standard Twain permet d'importer dans les logiciels d'applications des images provenant de scanners ou d'autres systèmes d'acqui-sition d'images. Pour le développement de ce standard, Lotus travaille en collaboration avec Eastman Kodak, Aldus, Caere, Hewlett-Packard et Logitech. L'objectif premier de ce groupe est de promouvoir l'utilisation de produits d'imagerie en développant une interface d'acquisition d'images intégrée et facile à employer. En intégrant cette technologie, Notes de Lotus permettra aux utilisateurs professionnels de saisir, de stocker et de partager des images de documents professionnels.

Lotus / Twain

• Nouveau standard d'acquisition d'images Twain • Support de ce standard par Notes de Lotus : logiciel de communication de groupe •

O_2 Technology - Formation

O_2 Technology assure initiation et formation aux systèmes de gestion de bases de données orientées-objet et au SGBD orienté-objet O_2

Cours d'un à quatre jours sur :

Les SGBD orientés-objet

Le système O_2, introduction ou formation approfondie

La programmation en C++ avec O_2

Les concepts étudiés sont immédiatement évalués sur stations de travail. Lieu de la formation : Versailles, ou à la demande, en entreprise. Renseignements et information : O_2 Technology Formation, 7 rue du parc de Clagny, 78035 Versailles Cédex. Téléphone : (33-1) 30 84 77 77. Télécopie : (33-1) 30 84 77 90. O_2 Technology est un organisme de formation agréé.

Re-sources

Commute

Edité et distribué par Central Point Software, Central Point Commute version 1.1 vient de sortir sur le marché. Logiciel de télétraitement, il fonctionne sous MS-DOS et permet la prise de contrôle à distance de n'importe quelle application.

Chaque micro-ordinateur sur lequel le logiciel est installé peut se comporter en poste maître ou en poste esclave. Fonctionnant entre deux PC à distance, par l'intermédiaire d'un modem, Central Point Commute peut être utilisé dans un environnement réseau local de type Novell NetWare et en mode direct par câble.

Commute

• Editeur : Central Point Software • Distributeur : Central Point Software • Logiciel de télétraitement • Fonctionne sous DOS et sous Windows (support de la souris à distance) • Compression de 50% en moyenne des fichiers • Sécurisé par mots de passe • Détection de 850 virus •

De plus, la prise de contrôle d'applications Windows est également possible. Et, dans ce cas, le support de la souris à distance peut aussi s'éffectuer. Ce logiciel à l'émission compresse les fichiers dans une proportion de 50%, diminuant ainsi les temps de connexion. Il intègre des fonctionnalités en termes de sécurité empêchant des accès illégaux sur le poste maître, des mots de passe à plusieurs niveaux permettant, par exemple, de consulter un fichier, mais pas de le rapatrier.

Central Point Commute détecte la présence de plus de 850 virus. Ce produit est vendu 1 290 F HT.

MKS : Lex & Yacc version 2.6

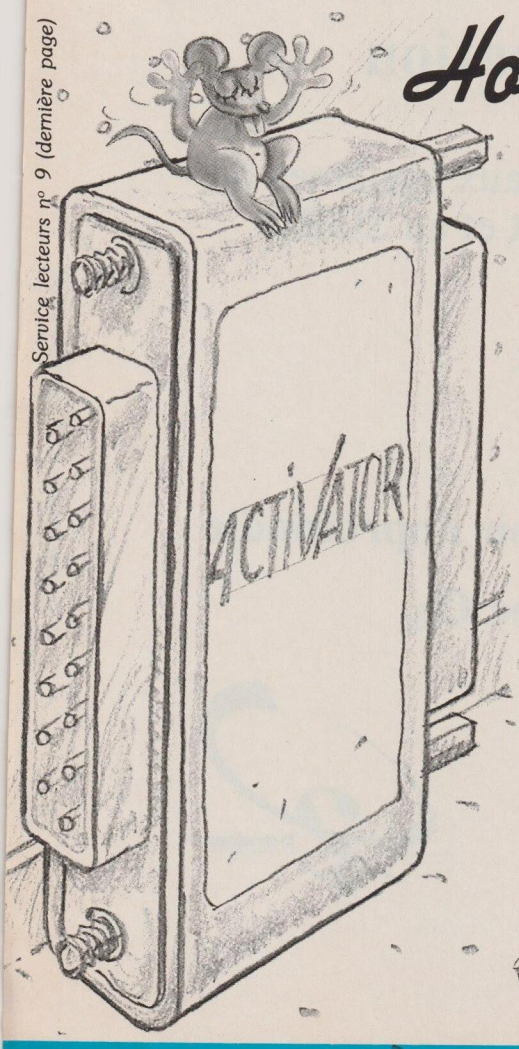
La nouvelle version de MKS Lex & Yacc supporte les langages C, C++ et pascal. Elle est également compatible avec le Phar Lap 286 Extender, permettant ainsi l'exploitation de la mémoire étendue. MKS Lex & Yacc est distribué par SienerSoft aux prix de 2 350 F HT pour la version OS/2 et de 3 200 F HT pour la version DOS et OS/2.

S-DOS version 1.31

Micropress commercialise la version 1.31 du noyau-temps réel compatible MS-DOS, S-DOS. Cette dernière mouture présente de meilleurs temps de réponse et intègre de nouvelles fonctions, notam-ment la gestion d'objets (un objet étant l'association d'un nom symbolique et d'un pointeur). La mise à jour d'une version antérieure est fixée à 200 F HT.

Service lecteurs n° 9 (dernière page)

Hors de portée des convoitises !!!



SÉLECTIONNÉE PAR **MICRO SIGMA** LA GAMME ACTIVATOR

REPOND A VOS BESOINS SPECIFIQUES DE PROTECTION INFORMATIQUE

- Contrôle d'accès à des programmes et/ou modules par mots de passe
- Contrôle du nombre d'utilisateurs pouvant simultanément accéder à un même programme
- Contrôles programmables à partir d'informations stockées dans une mémoire non volatile
- Environnements : DOS, UNIX, MAC, SUN...
- A partir de 250 Frs. H.T.

Consultez-nous au 46 22 99 88 Fax : 47 63 84 42

MICRO SIGMA Outils de Développement
42, rue Legendre - 75017 PARIS

ACTIVATOR est une marque déposée de SOFWARE SECURITY - USA

ACTIVATOR ACTIVATOR/N ACTIVATOR/S ACTIVATOR/M MACTIVATOR ACTIVATOR ACT

SUPERBASE 4 WINDOWS LUI A FOURNI UNE SOLUTION IMMEDIATE.

Afin d'aider ses directeurs régionaux à accéder instantanément à des informations urgentes en matière de bagages, vols et billets, la compagnie aérienne britannique a dû explorer de nouveaux horizons.

Elle a donc choisi la base de données relationnelle Superbase 4 Windows pour apporter une solution à un problème d'une complexité remarquable.

Les capacités graphiques de Superbase 4 fournissent des informations sous forme visuelle que les responsables de British Airways peuvent recevoir et enregistrer en un clin d'oeil, leur permettant ainsi de concentrer leurs activités non plus sur le traitement des données, mais sur de nouvelles méthodes de gestion.

Les développeurs d'applications pourront également s'étonner des avantages qu'apporte Superbase 4 en matière de productivité. Un gain de temps et d'argent pour tous, grâce à des instruments de programmation visuels et puissants qui les autorisent non seulement à concevoir des applications efficaces mais aident aussi les utilisateurs à se familiariser plus rapidement avec ces applications.

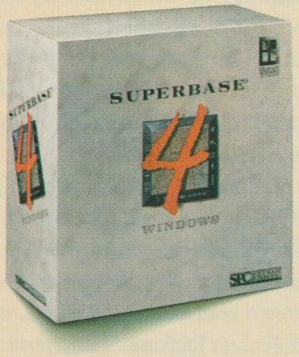
MICROSOFT WINDOWS

Bien entendu, un produit aussi sophistiqué est conçu pour grandir avec vous : le système est compatible avec les environnements existants, Superbase 4 est équipé d'une gestion de réseau souple et peut servir d'interface aux bases de données SQL.

Superbase 4, la base de données relationnelle Windows la plus vendue au monde est une solution qui a fait ses preuves dans de nombreuses entreprises leaders dans leur secteur.

Quand on sait que Superbase 4 est un produit de SPC, le 5ème éditeur mondial de logiciels micro, c'est un choix qui va de soi. D'autant plus que SPC est le concepteur de Harvard Graphics, le fameux logiciel de présentation graphique leader dans le monde.

A vous maintenant de découvrir Superbase 4 et de constater à quel point il pourrait transformer votre entreprise.



J'aimerais en savoir davantage sur la puissance de Superbase 4 Windows.
Je développe des applications pour d'autres utilisateurs. ☐ J'utilise des disquettes 3.5" ☐ 5.25" ☐

Mr ☐ Mme ☐ Mlle ☐ NOM _____ PRENOM _____
FONCTION _____
SOCIETE _____ NBRE SALARIES _____
ADRESSE _____
_____ CODE POSTAL _____
VILLE _____ TEL _____

FAXEZ LE COUPON AU **(1) 45 30 00 64** OU RETOURNEZ-LE À SPC FRANCE/ORSID CENTRE DE TRAITEMENT INFORMATIQUE, 6 RUE FIRMIN GILLOT, 75015 PARIS.

L et S 8

SPC SOFTWARE PUBLISHING CORPORATION

LE PARTENAIRE
INTELLIGENT DE VOS IDEES

Service lecteurs n° 10 (dernière page)

Re-sources

MS/3

Médiasys, la société française d'édition de logiciels, spécialisée dans les produits télématiques et télécoms, principalement sous OS/2, annonce aujourd'hui la disponibilité immédiate de son serveur de communication de troisième génération: MS/3. Celui-ci est un serveur vidéotex et un serveur de communication évolutif destiné aux PME/PMI et aux grands comptes.

MS/3

• Editeur : Médiasys • Distributeur : Médiasys • Serveur de communication à architecture répartie sous OS/2 • Version monoposte et réseau • Mise en place d'applications télématiques personnalisées •

Fonctionnant sur micro-ordinateurs PC/386 et 486, ce moniteur de communication existe en version monoposte ou en réseau et permet de mettre en place des applications télématiques personnalisées.

Pour ce faire, MS/3 supporte les terminaux Minitel, via X25, les réseaux téléphoniques commutés, la télécopie et les stations de travail (DOS, Windows et OS/2) en réseau local.

Doté de fonctions performantes, il garantit une intégrité et une fiabilité des applications assurées par les fonctionnalités OS/2, le support de plus de 500 accès simultanés en répartissant les applications télématiques sur plusieurs ordinateurs.

De plus, une architecture répartie lui permet la réalisation d'un nouveau système de communication dans l'entreprise, le but étant de mettre en relation un grand nombre d'applications avec une grande quantité d'utilisateurs. MS/3 pour une configuration de base, hors applications, est vendu 80 000 F HT.

Blast Professional

Cosyda présente les nouvelles versions de la gamme Blast Professional pour systèmes SCO Unix et Xenix, Interactive Unix, MS-DOS, et AT&T system V.

Cet ensemble est complété par une version esclave Bhost pour MS-DOS. Rappelons que Blast est un logiciel de communication asynchrone multi plate-forme full-duplex fondé sur une technique de déplacement de fenêtre, autorisant un transfert de fichiers à haut débit avec compression de données.

Il est équipé d'un langage de programmation et se destine à remplir des fonctions de transfert de fichiers, d'émulation de terminaux et de télémaintenance pour systèmes DOS et Unix à base de 386 et de 486. Blast supporte les protocoles standards du marché, tels que ZModem, Kermit, YModem, XModem, Checksum, etc.

Blast Professional

• Editeur : US Robotics • Distributeur : Cosyda • Logiciel de télécommunication et de télémaintenance informatique pour systèmes Unix et MS-DOS • Protocole full-duplex autorisant un transfert à haut débit avec compression de données • Emule la plupart des terminaux • Possède son propre langage de programmation orienté communication •

Les principales fonctionnalités de ces versions autorisent la prise en compte d'un PC distant pour des opérations de maintenance, de formation ou de transfert de fichiers et permettent un contrôle complet de l'écran et du clavier distants.

Blast supporte les écrans CGA, MCGA en mode texte et graphique avec conversion automatique entre les différents modes, un échantillonnage paramétrable pour le transfert rapide des écrans en mode graphique, la protection d'accès par mot de passe, et l'émulation complète d'une vaste gamme de terminaux. Le langage de programmation est de type basic orienté télécommunication.

Un nouveau serveur de réseau chez ALR

Infodis annonce la commercialisation de l'ALR Flyer LCT. Ce tower dispose de 12 emplacements pour disques et de 12 slots d'extension, dont deux 32 bits (les dix autres sont de type ISA 16 bits). Cette machine est proposée avec 4 Mo de mémoire vive (extensibles à 64 Mo) et avec des disques de 340 ou 535 Mo IDE.

Le contrôleur d'écran est de type S VGA et autorise une résolution de 1 024 x 768 points en 16 ou 256 couleurs (à condition d'acheter le module 512 Ko optionnel de mémoire vidéo).

Le Bios, d'origine Phoenix, est stocké sur une Flash ROM. Le modèle équipé d'un 486DX/33 coûte environ 30 000 F HT (40 000 F HT pour un 486DX2/50).

Fide

Le groupe des Technologies microprocesseurs et mémoires de Motorola (Austin, Texas) a présenté le 9 mars dernier, lors de la conférence de l'IEEE sur la logique floue, le Fide, nouvel outil logiciel d'environnement de développement d'inférence floue, développé par Aptronix, société de développement de logique floue.

Fide

• Introduction sur le marché de la logique floue • Un outil logiciel adapté a cette technologie : le Fide • Portage sur microcontrôleurs 68HC05, 68HC11, 68300 et DSP 56000 •

Le Fide représente la première partie d'un programme destiné à intro-duire sur le marché la logique floue et les technologies correspondantes.

Fide comporte des outils logiciels permettant aux experts en applications de donner aux microcontrôleurs standards des possibilités de logique floue. Au départ, ces dernières seront données aux familles de microcontrôleurs 68HC05 et 68HC11, puis ultérieurement au 68300 et, enfin, aux familles de processeurs numériques de signal (DSP) 56000.

Fide, outil logiciel, permet à l'utilisateur de concevoir de simuler un système dynamique complet, y compris le micro contrôleur spécifique, et d'évaluer facilement différentes solutions pour des aplications. Il assiste les développeurs dans la conception, le développement, le test, la mise au point et la génération du code système.

Fabian Pallini

Microsoft C/C ++ 7.0
Avec le plus complet des systèmes de développement orienté-objet, vous ne risquez pas de perdre le Nord.

PROGRAMMEZ AVEC CEUX QUI ONT LE PLUS D'EXPÉRIENCE SUR WINDOWS.

Avec Microsoft C/C++ 7.0, vous allez aborder la programmation orientée objet C++ et le développement d'applications pour Microsoft Windows dans les meilleures conditions.

Grâce à une implémentation rigoureuse du C++ 2.1 accompagnée des meilleures techniques d'optimisation, et à une documentation exhaustive, vous serez guidés pas à pas dans votre migration du C au C++. Pour bénéficier des avantages de la programmation orientée objet, vous exploiterez les Microsoft Foundation Classes. Rapides et économes en mémoire, ces classes facilitent le développement d'applications MS-DOS et Windows. Elles vous garantissent surtout la pérennité de votre code vers les futures versions 32 bits de Windows.

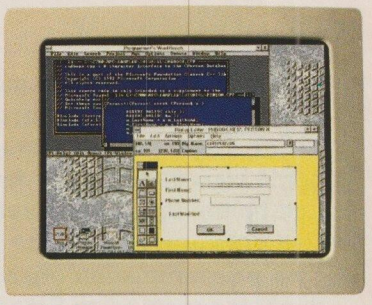
Microsoft C/C++ 7.0 est le plus complet des systèmes de développement C et C++ pour MS-DOS et Microsoft Windows.

- Installation de l'environnement à partir de Microsoft Windows.
- Un compilateur compatible 100% C ANSI et C++ ATT 2.1.
- Assembleur en ligne dans le code C et C++.
- Les meilleures technologies d'optimisation sur PC.
- Pré-compilation des en-têtes et des types, quel que soit leur contenu.
- Le packCode, pour diminuer de plus de 40% la taille du code.
- Les classes Microsoft Foundation Classes avec leur source.
- Tous les outils du Windows SDK 3.1 y compris le support de la programmation d'applications Multimédia, des extensions Windows for Pen Computing et d'Object Linking and Embedding.
- Des dizaines d'exemples de code Windows.
- L'environnement de développement Programmer's Workbench 2.0
- Plus de 4 Mo d'aide en ligne.
- Le Class-Browser, avec 14 vues différentes des données CodeView 4.0 fonctionnant dans une fenêtre Windows. Source profiler 1.2 pour analyser la performance de votre code.
- Le gestionnaire d'overlays MOVE.
- Le gestionnaire de mémoire virtuelle VM.
- Gratuit le gestionnaire de mémoire 388Max de Qualitas Software.

Et plus de 10 000 pages de documentation.

Son prix, seulement 3490 F HT !

Côté environnement de développement, il n'y a pas mieux que C/C++ 7.0 pour garantir votre productivité : vous ne quitterez jamais Windows dans toutes les phases de vos développements : le Programmer's Workbench et le débogueur CodeView s'exécutent parfaitement dans une fenêtre Windows ; et sous Windows, vos compilations peuvent s'exécuter en tâche de fond.



Enfin, si vous développez sous MS-DOS, vous ne serez pas en reste : le nouveau gestionnaire d'overlays Microsoft Overlay Virtual Environment va redonner une nouvelle jeunesse à vos applications qui vont pouvoir comporter encore plus de code (jusqu'à 16 Mo !) et même gérer de la mémoire virtuelle pour les données.

Dans tous les cas de figures, Microsoft C/C++ 7.0 est résolument l'outil de classe professionnelle. Il n'en est pas moins abordable puisqu'il vous est proposé au prix public conseillé de 3490 F HT. Et si vous possédez déjà un compilateur C ou C++, quelle que soit sa marque, nous vous proposons de l'échanger pour C/C++ 7.0 pour seulement 1290 F TTC (offre valable jusqu'au 31 juillet).

Pour toute information complémentaire :

3616 Microsoft

Le Service Clients : 16 (1) 69 29 11 11

Support Technique : 16 (1) 69 86 10 20

Microsoft : Infomart CNIT - La Défense

MICROSOFT - 91957 LES ULIS CEDEX

Microsoft®

LES LOGICIELS QUI DONNENT DES AILES.

Service lecteurs : n° 11 (dernière page)

ROUX SEGUELA CAYZAC & GOUDARD

SYSTÈME

LA STATION NEXT

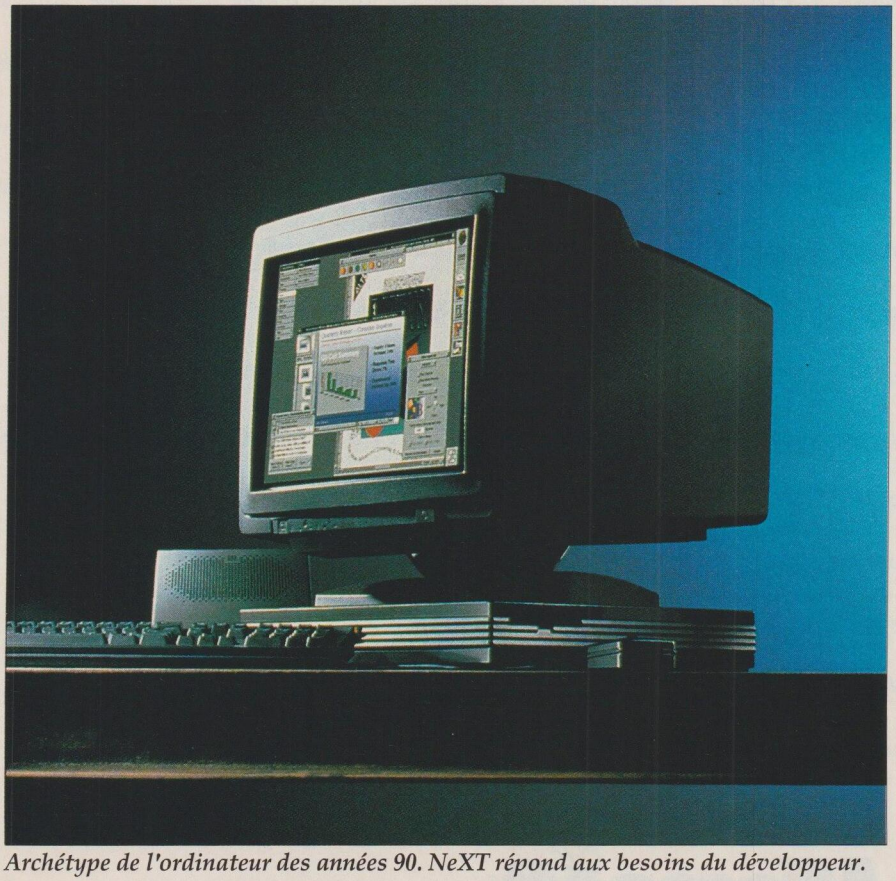
La plate-forme de développement idéale existe, nous l'avons testée

FABIAN PALLINI

La station NeXT reste méconnue de l'immense majorité des développeurs, en dépit de ses qualités. Celles-ci sont grandes, comme nous avons pu le vérifier par le biais de ses outils de développement qui gardent une bonne longueur d'avance vis-a-vis compilateurs C++ des autres stations de développement.

La programmation orientée objet (POO) a vu le jour dans les années 60 en Scandinavie et tend aujourd'hui à révolutionner le monde de la programmation structurée. Celle-ci, après avoir régné pendant une vingtaine d'années, laisse place de nos jours à la POO avec sa modélisation d'objets. En effet, face à la complexité et à la taille en moyens humains et matériels que nécessitent les projets informatiques, les outils de programmation doivent être performants pour que les développeurs puissent mener à terme les applications dans les meilleures conditions.

Plus l'ordinateur prend part au travail de conception, plus les programmes sont faciles à tester, à utiliser et à maintenir. C'est la raison pour laquelle que les langages orientés objet connaissent un tel essor, tant dans le domaine de l'intelligence artificielle que dans celui du génie logiciel. De plus, les interfaces graphiques qui rendent plus conviviale et plus intuitive l'utilisation de l'ordinateur se généralisent sur toutes les plates-formes existantes. La programmation et les outils de développement n'échappent pas à cette tendance, allant même jusqu'à la devancer. En 1990, la venue de la seconde génération de la gamme NeXT a suscité la curiosité des informaticiens.



Archétype de l'ordinateur des années 90. NeXT répond aux besoins du développeur.

Le NeXT se présente, en effet, comme l'ordinateur de cette décennie et innove en intégrant le multimédia en standard. Son système d'exploitation multitâche «Mach» est entièrement compatible avec Unix BSD 4.3, et son interface graphique NeXTStep apparaît totalement orientée objet.

Cette machine est certes conçue pour l'utilisateur, mais aussi pour rendre le travail du développeur rapide, modelable et confortable. Steve Jobs, fondateur des sociétés Apple et NeXT, dit à ce sujet que ce qu'Apple à fait dans les années 80 pour les utilisateurs, NeXT le fera dans les années 90 pour les développeurs. Mais si l'on compare la part de marché NeXT à celles du PC et du

Née avec une cuillère d'argent. Telle est la devise qui pourraît être appliquée à la station NeXT. Elle offre, en effet, des possibilités qui n'existent encore qu'à l'état embryonnaire sur les PC et présente une ergonomie à faire pâlir d'envie le plus convaincu des développeurs Mac. Le tout pour un prix raisonnable en regard de son équipement et de ses performances.

Mac, rares sont les développeurs qui connaissent le NeXT en tant qu'outil de développement. Il était donc d'actualité de présenter au lecteur la machine NeXT, les concepts de sa programmation et de faire le tour des outils de développement. Le matériel utilisé a été une station couleur NeXT à base d'un Motorola 68040 intégrant une unité de gestion de la mémoire (MMU), un coprocesseur arithmétique, un disque dur de 400 Mo, dont le temps d'accès est de 17 ms, 16 Mo de RAM et un écran couleur 17 pouces.

Premier contact

La machine a un design très étudié. Elle se compose de 5 éléments :

• l'unité centrale très peu encombrante (40 x 36 x 6 cm) qui contient la carte mère. Celle-ci comprend un circuit vidéo permettant un affichage couleur 16 bits par pixel et pouvant émuler la qualité 32 bits;

• un disque dur de 400 Mo ayant un temps d'accès de 17 ms et un lecteur de disquettes 2,8 Mo (disquettes ED) compatible 1,44 Mo et 720 Ko pour l'import ou l'export de fichiers au format DOS;

• un moniteur 17 pouces haute résolution (1 120 x 832 points) ayant une vitesse de rafraîchissement de 68 Hz en mode non entrelacé, ce qui procure un véritable confort visuel. La carte mère permet l'affichage de 4 096 couleurs simultanées;

• le clavier, à partir duquel on peut contrôler les fonctions physiques de l'ordinateur telles que la mise sous et hors tension du NeXT, le réglage de la luminosité de l'affichage et du volume du son. Regrettons que les touches Deb, Fin et Suppr soient absentes du clavier, elles rendent bien des services.

• le cinquième élément est la Sound Box (la boîte à musique) qui intègre un haut-parleur et un microphone grâce auquel on peut enregistrer des messages vocaux ainsi que de la musique. Par ce biais, l'échantillonnage se fait sur 8 bits en code Mulaw qui présente une forme

Programmes contenus dans la version 2.0 de NeXTStep

NeXTStep comprend les programmes suivants :

• Workspace Manager : correspond au finder sur Mac et au gestionnaire de programmes sous Windows.
• NeXT Mail : programme E-Mail.
• Write Now : traitement de texte.
• Digital Webster : dictionnaire (anglais).
• Thesaurus : dictionnaire des synonymes.
• Edit : éditeur de programmes.
• Mathematica : seulement pour les universités.
• DataViz/Bridge : passerelle pour le transfert de données vers Mac.
• Installer : installation de logiciels.
• FaxReader : travail avec les fax-modems.
• Preferences : spécifications internationales du système d'exploitation.
• PreView : représentation des dessins PostScript.
• PrintManager : gestionnaire d'imprimantes.
• VT100 Terminal : émulateur de terminal VT100.
• BuildDisk : formatage des disques durs.
• MailManager : gestionnaire de mail.
• NetManager : gestionnaire de réseau.
• UserManager : gestionnaire de compte utilisateur.

La version développeurs contient en plus les programmes suivants :

• Interface Builder : système de développement graphique.
• Objective-C : compilateur objective-C.
• C++ : compilateur C++.
• Classes Objective.
• C : définition des classes.
• 56001 DSP Tools : utilitaires pour le processeur de traitement signal.
• GNU Emacs : éditeur Emacs.
• PostScript Tools : outils pour le travail en PostScript.
• Application Kit : kit d'application.
• Music Kit : classe d'Objective-C.

de compression correspondant à un code 16 bits;

Quant aux connexions, elles sont faciles d'accès et rationnelles. Elles se composent d'une prise mâle 220 V d'où sortent deux câbles d'alimentation, un pour l' UC et l'autre pour le moniteur. Un câble en Y permet de connecter l'UC à l'écran et à la Sound Box. Enfin, un câble relie le clavier à la Sound Box et un autre joint la souris au clavier. C'est bien calculé.

Des possibilités étendues de connexions

Derrière l'UC, un port SCSI permet l'adjonction d'une mémoire de masse de tout type. Deux ports série A et B, à la norme RS423 (compatible RS232C), permettent la connexion de modems, de fax et d'imprimantes PostScript. La vitesse maximale de transmission est de 38 400 bauds.

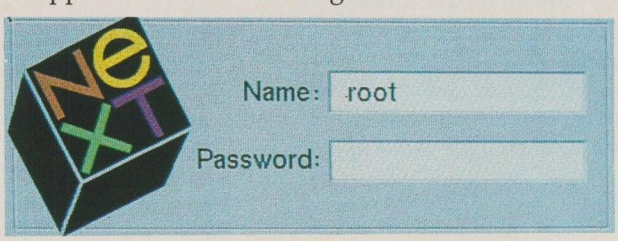
Un port DSP (Digital Signal Processing) donne accès au processeur de traitement de signal numérique du NeXT (le DSP Motorola 56001 cadencé à 25 MHz). La numérisation par le DSP s'avère parfaite, d'une qualité CD. Un port imprimante non standard autorise la connexion de l'imprimante laser NeXT d'une résolution de 400 ppp.

On regrettera que la version 2 de NeXTStep ne supporte pas l'impression couleur sans l'ajout de logiciels commerciaux. La station NeXT offre, par ailleurs, des possibilités de connexions à un réseau Ethernet de type Thin-Ethernet ou Twisted-Pair-Ethernet, grâce à un connecteur pour paire torsadée et un autre pour câble coaxial.

Quant à la Sound Box, elle incorpore une sortie casque, et deux prises cinch en sortie permettent de connecter un amplificateur stéréo. Enfin, une prise femelle jack est disponible pour un microphone externe.

La mise sous tension

Pour entrer dans l'univers NeXT, il suffit d'enfoncer le bouton Power du clavier. Mach, le système d'exploitation compatible Unix 4.3, charge les programmes de l'interface NeXTStep, à savoir, le Windows Server et ses fonctions de Display PostScript ainsi que le Workspace Manager (l'espace de travail).
Puis apparaît la fenêtre de «login» :



DU NOUVEAU CHEZ NEXT

NeXT sort la version 3 de NeXTStep, dont les principales améliorations sont :

❶ Le DBKit orienté objet pour rendre possible le développement d'applications accédant à des bases de données en tirant totalement avantage de l'environnement NeXTStep.
❷ Le support des réseaux Novell et AppleTalk, permettant aux utilisateurs de partager des données avec des ordinateurs sous DOS, OS/2, Windows, MacOS et Unix.
❸ L'intégration au sein d'Objective-C des possibilités d'envois de messages à des objets sur d'autres stations ou sur d'autres applications. Une nouvelle génération d'applications de travail de groupe est donc rendue possible.
❹ Un système de liaison dynamique entre applications NeXTlink qui permet aux groupes de travail de partager l'information de façon dynamique.
❺ Le Phone Kit constituant des classes d'objets pour l'interfaçage du téléphone analogique et pour la transmission de données à grande vitesse à travers un réseau RNIS.
❻ Le 3D graphics Kit qui permet d'ajouter des graphiques 3D à des applications.

A NOTER

NeXT a passé un contrat avec la société Sorbus. Celle-ci assurera la maintenance gratuite pendant un an de chaque machine NeXT vendue. Pour l'année suivante le contrat de maintenance s'élève à 10% du prix d'achat. Il existe deux styles de formation sur NeXT, l'une réalisée par la société mère (orientée programmation et utilisation de l'IB), l'autre proposée par deux sociétés : Qualigraf et El Camino Real (utilisation de bases de données).
X Window a été porté sous NeXT, notamment par la société française Cub'x.
La version portée est X en version 11 release 5.
Il existe une messagerie des utilisateurs NeXT en France et dans le monde. Son nom est Olean Cean.
Mach, le système d'exploitation compatible Unix de NeXT, a été retenu par l'OSF (Open Software Foundation) pour devenir un standard.

Après l'entrée du mot de passe, apparaît le Workspace Manager qui donne accès aux répertoires, aux fichiers et aux applications (*Cf. l'article sur NeXTStep de ce numéro*). Il se compose de trois éléments :
• Le menu principal avec ses sous-menus hiérarchiques détachables qui permet de configurer l'environnement selon ses préférences. On y crée des répertoires, y vide le contenu du Recycler (équivalent de la corbeille sous Mac), y formate une disquette... Il autorise le choix du style de vue des fichiers (browser, icône ou listing «à la Unix».) et inclut des utilitaires comme l'Inspecteur de

fichiers, la recherche de fichiers, la visualisation des process actifs, etc. Le menu du Workspace Manager offre en plus les services de librairie pour les recherches dans un fichier ou dans une quelconque documentation ainsi que le service mail pour la communication inter-utilisateurs via le réseau.

Comme dans toutes les applications, le menu principal contient la commande Hide. Le système étant multitâche, plusieurs applications peuvent être visibles à l'écran : il est possible de les cacher sans pour cela les désactiver. C'est également par le menu du Workspace Manager que l'on se «délogue».

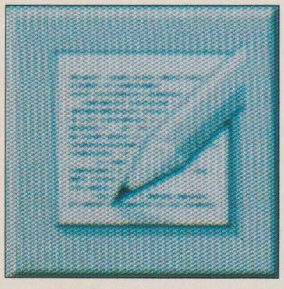
• Le File Viewer (fenêtre des fichiers) qui se compose de 3 parties. En haut, l'étagère où les répertoires dont on se sert souvent sont rangés, ce qui évite la recherche dans l'arborescence du disque. C'est aussi une zone de transit pour les répertoires et pour les fichiers à manipuler. En bas, le browser montre le contenu des répertoires du disque sur une branche.

En double-cliquant sur un fichier ou sur la sélection de plusieurs fichiers, les applications s'y rapportant sont lancées (selon le procédé OLE). Enfin, au milieu, le chemin d'accès est représenté sous forme d'icônes que l'on manipule pour copier, entre autres, les fichiers et les répertoires.

• L' Application Dock (entrepôt d'applications) qui se présente sous la forme d'une colonne d'icônes. Seule l'icône NeXT (du Workspace Manager) et celle du Recycler font toujours partie de l' Application Dock.

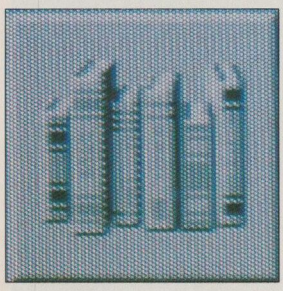
Les autres y sont déposées par l'utilisateur en fonction de ses besoins. Cette opération s'effectue en draguant l'icône d'une application active du Workspace Manager vers l'Application Dock. Pour lancer une application, il suffit de double-cliquer dessus.

Développer avec NeXT

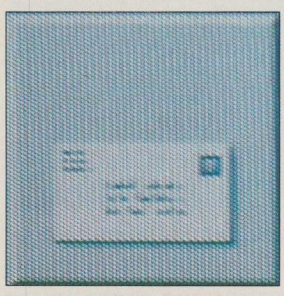


L'application Edit permet l'édition de programmes, quel que soit le compilateur utilisé (Objective-C, C++, fortran, lisp, pascal...), et aussi des procédures Shell d'Unix. Le menu de l'éditeur dispose de tous les outils propices à la mise en page (polices de caractères extensibles notamment avec le Plus Pack d'Adobe) et des opérations de copier-coller via le presse-papiers pour l'import de texte entre applications.

Dans l'optique d'aide au développement pour la compréhension des programmes, notons la possibilité de rendre momentanément invisible une ou toutes les portions de codes entre accolades. Edit donne, de plus, accès aux services tels que le dictionnaire et celui des synonymes, le mail ainsi que le Librarian.



Le disque dur comprend 25 Mo de documentation. Le Librarian accède aux données symbolisées sous forme de livres posés sur son étagère. Par défaut, il charge les principales documentations de NeXT, ce qui permet, dans un minimum de temps, d'atteindre le sujet désiré ou l'exemple dans lequel la fonction cherchée apparaît. La recherche se faisant par indexation, elle est instantanée. Tout comme dans les dictionnaires, on peut y rajouter d'autres livres, même les siens!

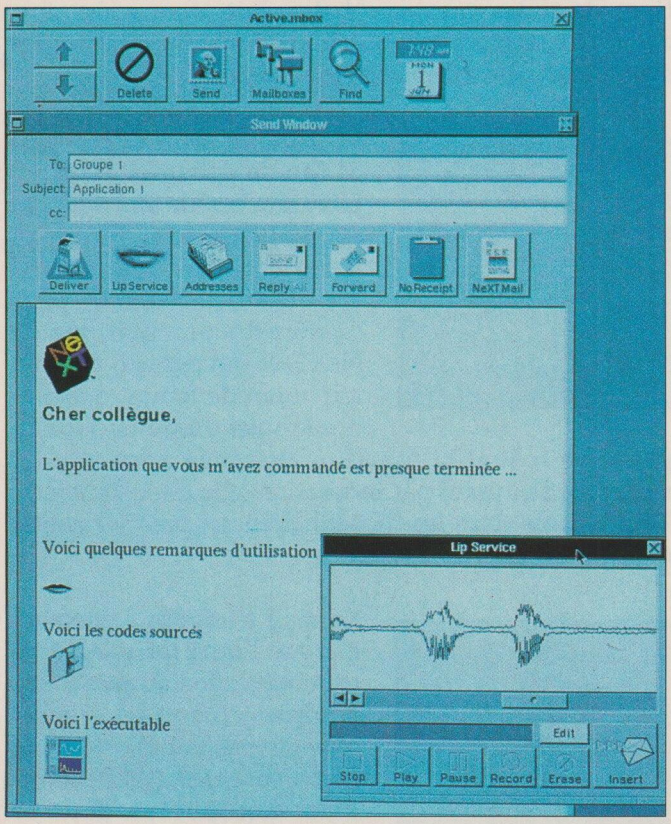


Le mail s'impose également comme l'outil privilégié de communication au sein d'une équipe. Jusqu'ici, nous n'avons pas fait allusion aux nombreuses facultés multimédias incorporées dans la machine NeXT. Le mail en est l'un des exemples les plus frappants. En effet, en sus du texte, son, paroles, images, icônes, fichiers, répertoires et exécutables peuvent figurer au sein d'un mail. Le ou les destinataires équipés également d'un NeXT, répertoriés dans le carnet d'adresses, recevront dans leur mail toutes les données concernant les composantes d'un projet (*voir photo page suivante*). Le NeXT est, à l'heure actuelle, le système ouvert par excellence, d'une part, en raison de son système d'exploitation à base d'Unix, d'autre part, grâce à ses possibilités de connexion à un réseau hétérogène. Il peut transmettre des mails au format ASCII aux différentes machines qui s'y trouvent connectées.

Un langage de POO : Objective-C

Dans la tradition Unix, le système d'exploitation Mach est entièrement écrit en C. Ce langage bénéficie d'une couche objet, offrant ainsi aux programmeurs la puissance du langage C au sein d'un environnement objet.

Toutes les composantes du système sont divisées en classes, lesquelles ne constituent pas à proprement parler des objets. En Objective-C, la classe est la description des champs et des méthodes et des informations communes à un ensemble d'objets similaires. La racine de l'arbre d'héritage correspond à la classe objet. Il faut



Sons, images, fichiers et répertoires peuvent intégrer un mail.

savoir que les développeurs peuvent créer leurs propres classes d'objets dans l'arbre d'héritage. Ce dernier est donc calqué sur celui des classes conçu par le programmeur. Les prototypes des classes de cet arbre sont des instances. Un objet constitue l'instance de sa classe, ses données ont un caractère privé et ses méthodes (procédures internes à l'objet) effectuent les opérations qui lui sont propres. Elles peuvent également contenir des envois de messages. En effet, les objets communiquent entre eux par des messages. Un message invoque une méthode propre à l'objet. La syntaxe d'un message est :

```
[ObjetReceveurNomDuMessage:Argument1,Argument2];
```

Les classes d'Objective-C sont déclarées et définies par deux fichiers : celui d'interface (.h) et celui d'implémentation (.m).

NeXTStep représente l'interface d'environnement graphique pour les programmes d'applications.

Le Window Server

Le NeXT Window Server est le système qui administre les périphériques d'interface, la souris, le clavier et l'écran, dont l'affichage intègre complètement le langage PostScript. A ce sujet, l'utilisation de PostScript à travers le NeXT offre deux avantages : l'impression et l'affichage sont homogènes. En effet, l'information affichée correspond à celle imprimée sans conversion (Wyswyg).

Il donne la liberté de déplacer du texte formaté et des graphiques entre applications.

Le process Window Server tourne constamment en tâche de fond. C'est lui qui supporte l'affichage des applications.

L'Application Kit

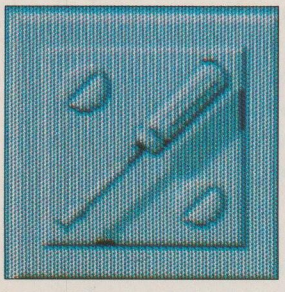
L'Application Kit constitue une librairie de classes et de routines C utilisées pour implémenter toutes les applications dans l'environnement NeXTStep.

Il est la couche objet de NeXTStep dont bénéficie de façon homogène chaque application.

On peut créer des instances de classes déjà définies dans l'Application Kit (fenêtres, boutons, menus, et bien d'autres composantes.) ainsi que définir à son tour des sous-parties des classes de l'Application Kit. Ces dernières, en plus de leurs propres fonctionnalités, héritent des mêmes caractéristiques que leur classe mère. Au sommet de l'arborescence de l'Application Kit, il y a la classe objets.

En créant une sous-classe d'objets, on intègre les objets propres à l'application. Notons, qu'une fois créés, les objets sont réutilisables dans n'importe quelle application. L'Application Kit lui délègue alors la responsabilité de traiter ses propres messages. Dans la plupart des cas, une application est vouée à répondre aux événements du Window Server concernant sa fenêtre. La classe objets remplit quatre tâches. Elle reçoit les événements du Window Server et les distribue vers les autres objets.

Elle administre les objets fenêtre et obtient par défaut les messages de l'objet Listener qui intervient, avec l'objet Speaker, dans la communication inter-process. Elle garde, enfin, les informations globales partagées par les autres objets.



L'Interface Builder (IB) est l'outil graphique de développement privilégié de NeXT. Cet outil, conçu par un Français, Jean-Marie Hulot, est davantage qu'une boîte à outils graphique. En effet, l'IB exploite les caractéristiques de lien dynamique permettant à des objets d'être reliés, sans compilation. On peut donc construire l'interface d'une application sans perdre de temps dans la boucle

compilation, édition, test. L'Interface Builder gère également le management des fichiers intervenant dans le projet (codes source (.h et .m), les librairies d'objets (autres que l'Application Kit), images, icones, sons, sous-projets.

Des bouts de code source C , C++(Ansi) ou fortran peuvent également faire partie du projet. L'Interface Builder s'occupe aussi du fichier Make File et génère le squelette du code source en Objective-C. Il ne reste qu'à produire le code manquant et à lancer la compilation.

Pour illustrer le développement avec l'Interface Builder, nous avons mis au point une petite application ayant pour but le calcul de la TVA avec des taux variants qui comprend une zone de saisie, deux zones d'affichage, un bouton de calcul, quatre boutons radio, un panneau d'information et un menu permettant notamment l'impression. On intègre au projet un objet calculatrice déjà existant. Et, pour finir, on dote l'application d'une icône.

Commencer un projet

Il faut, en premier lieu, charger l'IB en double-cliquant sur son icône dans l'Application Dock, puis sélectionner New du menu File, le sauvegarder dans un répertoire de travail et cliquer sur Projet du menu File.

A ce stade, apparaissent six éléments (*voir photo ci-dessous*). Le menu principal d'IB, avec son sous-menu File, comprend entre autres les commandes Test, pour l'interface, et Make, pour produire en fin de parcours l'exécutable.

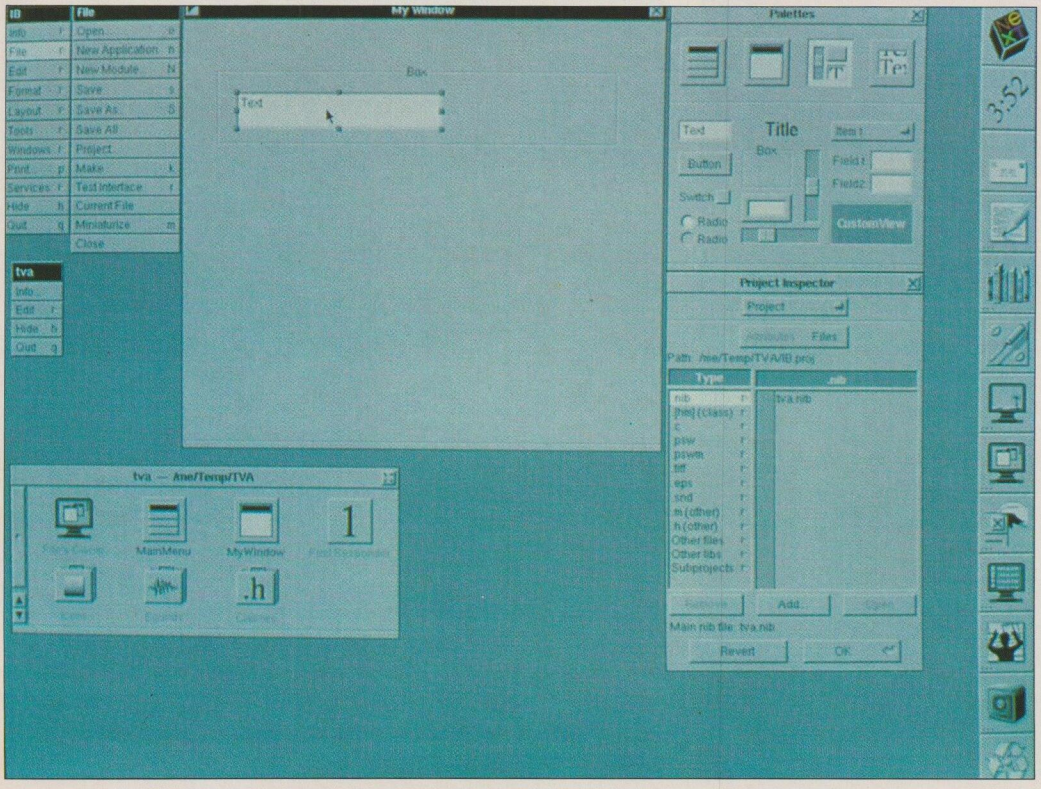
Ce menu intègre également les commandes de mise en page des objets (centrage, alignement...). La fenêtre des palettes est localisée en haut à gauche.

Quatre palettes d'origine sont à la disposition du développeur :

- la palette des objets classiques (boutons, potentiomètre, switch, menu pop-up)...;
- la palette des menus standards intégrant déjà leurs propres fonctionnalités;
- une palette permettant l'ajout de nouvelles fenêtres au sein du projet;
- une palette pour les zones de texte et les fenêtres de gestion de fichiers.

En bas à droite, l'Inspecteur donne accès aux différents éléments du projet : la liste des fichiers qui le composent, les informations sur les classes, les connexions graphiques des objets, les variables d'instances et des méthodes relatives à un objet. S'ajoute tout ce qui est nécessaire aux attributs des objets de l'interface (redimension d'une fenêtre, position du texte, emplacement de l'icône dans un bouton...).

En bas à gauche se situent la fenêtre des fichiers contenant les objets de l'Application Kit dont hérite par défaut l'application ainsi que les trois valises



L'Interface Builder prend en charge le fichier Make File et génère le squelette du code source.

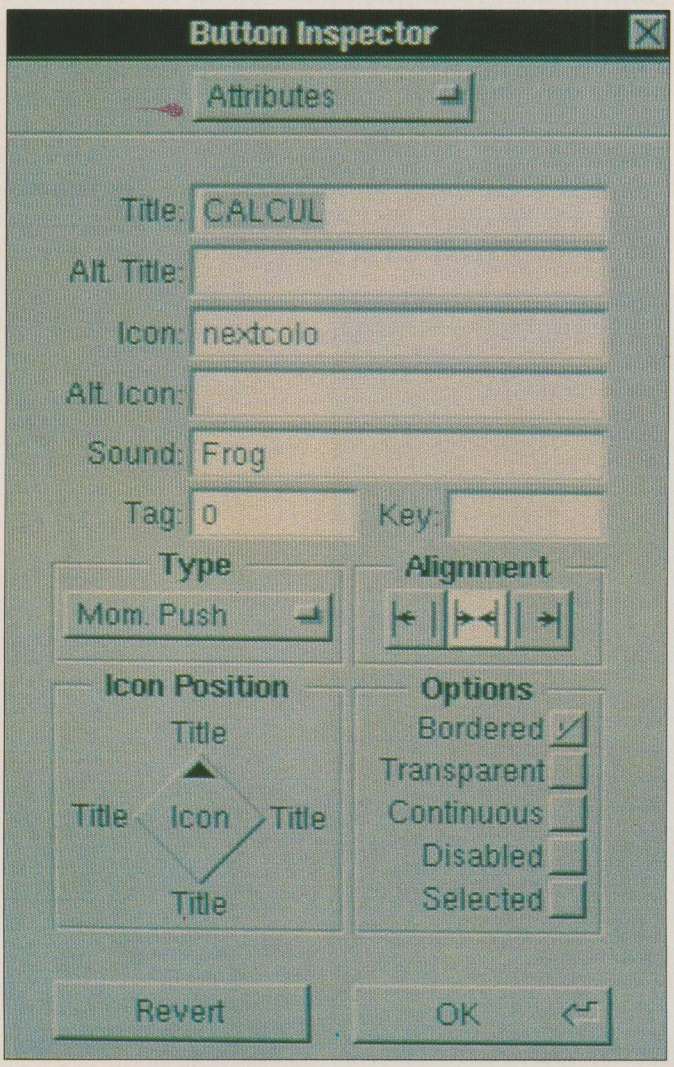
des sons, des icônes et des classes correspondant aux ressources du projet.

Construire l'interface

Pour importer des objets dans une fenêtre, il suffit de draguer les outils de la palette dans celle-là, de les travailler dans cette dernière, puis d'en définir les caractéristiques à l'aide de l'Inspecteur qui se modèle selon le type d'objet sélectionné. Par exemple, pour le bouton Calcul, on drague une boîte (Box) qui sert de cadre au bouton, puis, avec le menu Attribut de l'Inspecteur, on choisit le type de cadre, l'option sans titre.

On drague alors de même un bouton, que l'on pose sur le cadre et, après l'avoir dimensionné et centré, on remplace le mot Bouton par le mot Calcul. A l'aide du panneau de polices de caractères on change alors sa taille, sa police et son style. Si l'on décide de mettre une icône sur le bouton, on ouvre la valise des icônes dans laquelle on avait, en plus de celles d'origine, importé le sigle de NeXT en fichier PostScript. Toujours sur le même principe, il suffit de draguer l'icône de sa fenêtre sur le bouton et de choisir son emplacement par rapport à Calcul dans l'Inspecteur (*voir photo de gauche*).

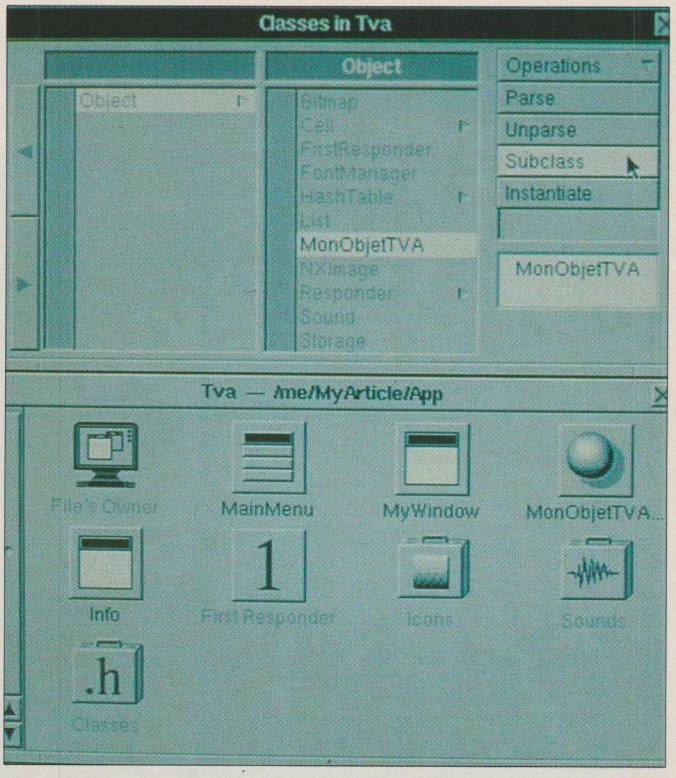
Notons qu'avec la même démarche, on aurait également pu adjoindre au bouton Calcul un son. On se serait alors servi de la mallette des sons. Chaque objet se sélectionne et se déplace à l'aide de la souris dans un espace 3D (arrière-plan, premier plan). A l'aide de la touche Alternate maintenue pendant l'agrandissement d'un objet, ce dernier se duplique. On crée ainsi des matrices d'objets. C'est avec cette technique que l'on a réalisé les 4 boutons radio des taux. Puis, on les a renommés.



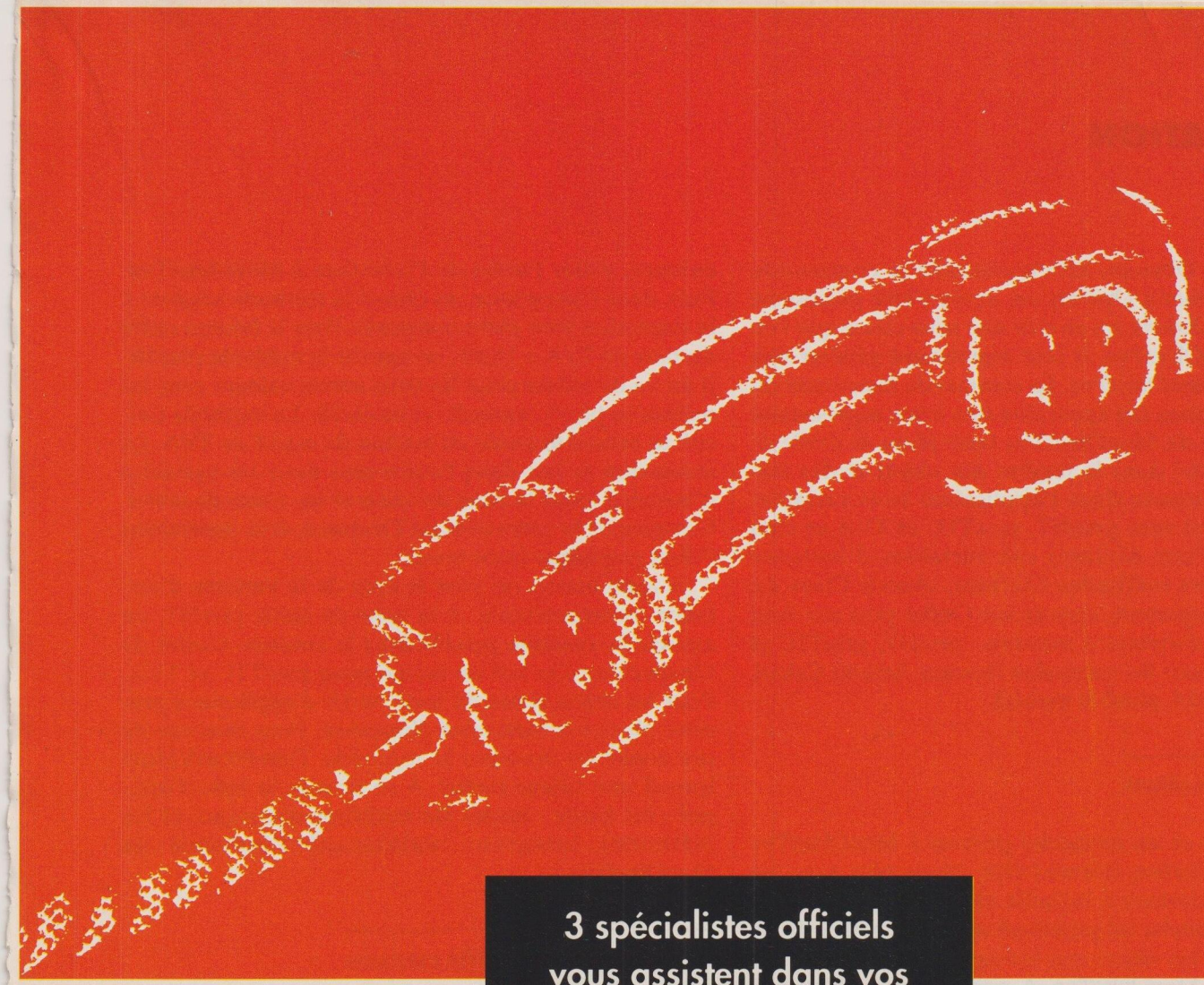
Modification des attributs des éléments graphiques composant l'application.

Créer l'objet TVA

Une fois l'interface graphique réalisée, il faut créer la classe de l'objet. Pour ce faire, on ouvre la valise des classes. Apparaît alors à l'écran une fenêtre représentant l'arbre des classes du kit d'application. Notre objet étant un objet à part entière, il doit être une sous-partie



Création de sous-classe dans l'arbre d'héritage et instanciation d'objets.



3 spécialistes officiels vous assistent dans vos développements Novell.

Développeurs, décrochez le téléphone rouge.

Vous n'êtes jamais seul avec un outil de développement Novell.
Dans le cadre de son "Professional Developer's Program", Novell vous garantit l'assistance totale de trois Importateurs Officiels. Vous pouvez tout leur demander.
Assistance logistique, conseil et équipement rapide direct pour développer sous Novell.
Assistance par la formation qualifiée des développeurs.

Notez leurs noms à l'encre rouge.

Alfratronic
ZI de Courtabœuf
7 av. du Canada - B.P. 310
91958 Les Ulis Cédex
Tél. 69 86 75 00

Interquad
64-66, rue Escudier
92100 Boulogne
Tél. 46 84 05 15

Omnilogic
11, rue de Cambrai - Bât. 28
75019 Paris
Tél. 40 05 28 00

NOVELL

Le passé, le présent, l'avenir de l'informatique réseau

HARD communication

Service lecteurs n° 12 (dernière page)

de la classe Objects. En choisissant la commande Subclass du menu Operations, après s'être positionné sur la classe Objects, on en crée une sous-classe que l'on nomme MonObjetTva. Il faut maintenant générer une instance de MonObjetTva. La commande Instantiate du menu Operations fait apparaître une icône représentant une instance de MonObjetTva (*voir photo de droite en page précédente*) dans la fenêtre des objets et des ressources. On trouve dans l'application trois variables d'instance qui sont prixHT, prixTTC et TVA et cinq méthodes. Pour déclarer ces variables d'instance (Outlet) et ces méthodes (Action), choisir Class du menu de l'Inspecteur, cliquer sur Outlet puis sur Add et entrer les variables d'instance : prixHT, prixTTC et TVA. Idem pour les méthodes (bouton Action) Calcul, plus18, moins18, plus5 et moins5.

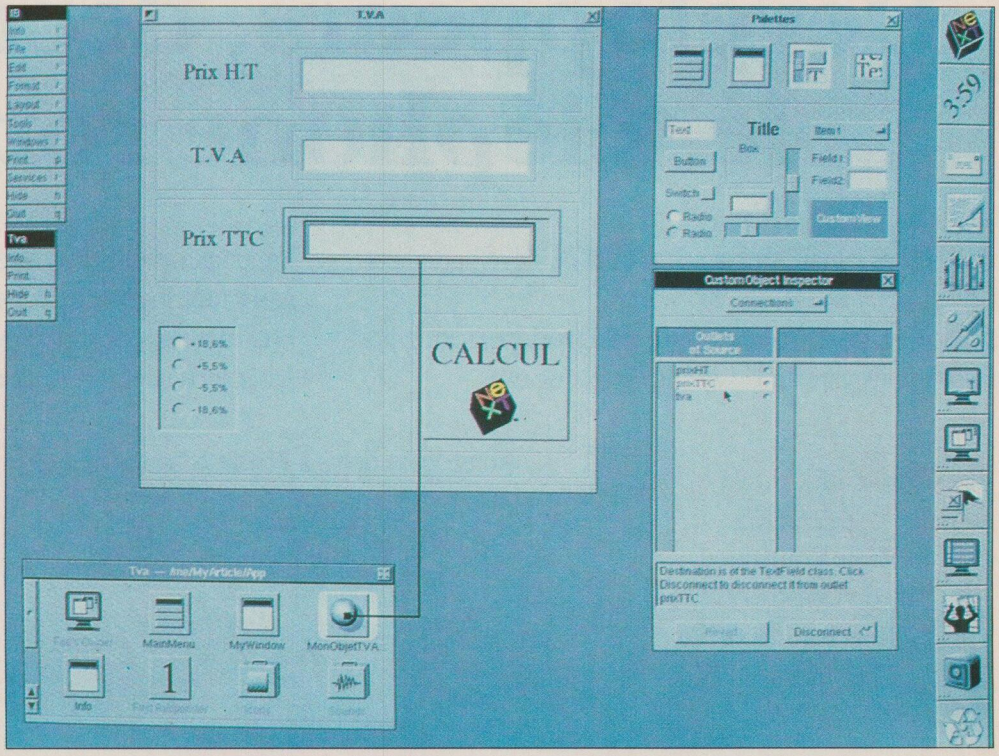
Connecter les objets

Les variables et les méthodes propres à MonObjetTva une fois déclarées, doivent être connectées avec les objets de l'interface à laquelle elles sont directement liées. La démarche est purement mécanique. Il suffit de joindre tous les objets qui invoquent une méthode à l'instance de MonObjetTva. A cette liaison, on associe la méthode correspondante. Pour établir cette connexion, il faut maintenir la touche Control du clavier enfoncée, pointer l'objet qui doit invoquer une méthode et, tout en gardant le bouton de la souris enfoncé, relier l'objet de l'interface à MonObjetTva. Puis, dans l'Inspecteur, on choisit la méthode adjointe à la connexion (*voir photo ci-dessous*). Pour le bouton radio du taux de TVA = 18,6%, on associe la méthode plus18:. Il en va de même pour les trois autres valeurs avec les méthodes plus5:, moins5: et moins18:. La méthode Calcul: se trouve associée au bouton de calcul de TVA. La connexion se fait dans l'autre sens pour les variables d'instance, pour la zone de saisie et les zones de sortie. L'origine de la connexion est l'instance, et la destination l'objet graphique (zone texte).

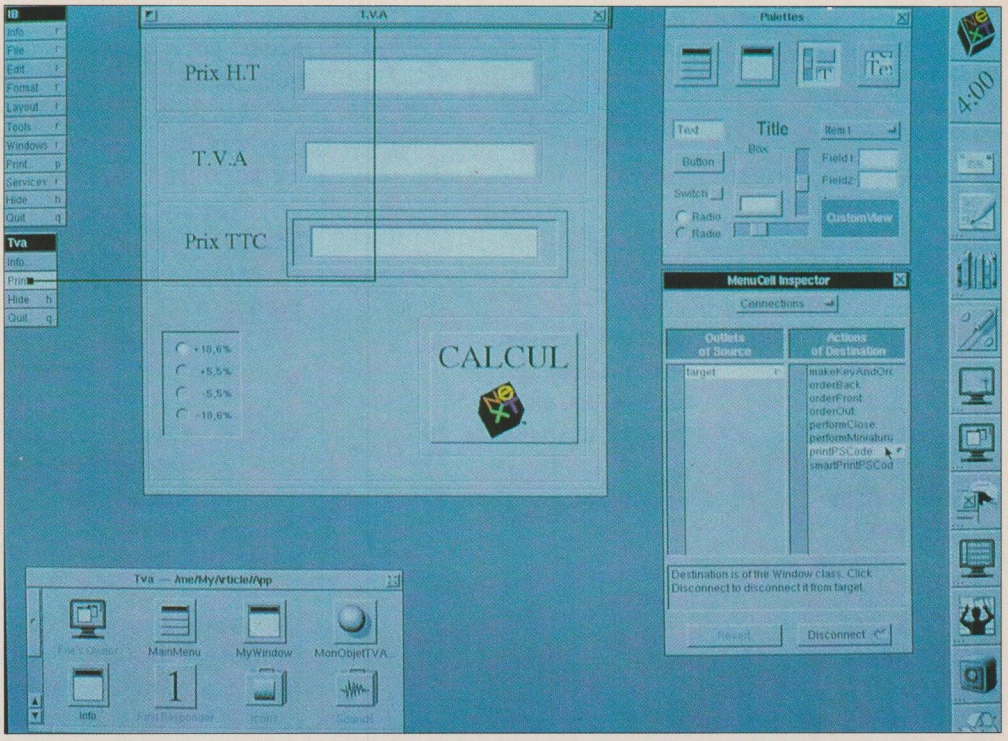
A cette liaison on joint le nom de la variable d'instance correspondante. A la zone de saisie prixHT, on associe la variable prixHTdans la colonne Outlet de l'Inspecteur. On fait de même pour prixTTC et TVA. Pour les objets zones de texte, le kit d'application propose plusieurs méthodes d'instance. Pour chacune de ces zones on invoque la méthode TakeFloatValue: (prend une valeur réelle). A ce niveau, la toile d'araignée est tissée : toutes les variables d'instance et les méthodes intègrent maintenant l'instance de MonObjetTva.

Intégration d'objets déjà existants

Nous avons ajouté à l'application un objet calculatrice importé d'une palette préalablement chargée dans la fenêtre des palettes. Après avoir dragué la calculatrice à l'intérieur de la fenêtre TVA, on a lié la calculatrice à la



les connexions entre les variables et les méthodes avec les objets s'effectuent uniquement de manière graphique.



Une application s'apparente à un jeu de construction. Par exemple, l'option d'impression s'obtient en invoquant la méthode PrintPSCode

zone de saisie du prixHT et associé la méthode prédéfinie TakeFloatValue:. Le résultat de l'opération effectuée par la calculatrice est, par conséquent, reporté dans l'objet zone de saisie du prixHT. Nous avons également créé un panneau d'information à l'aide de la palette des fenêtres et l'avons connecté au menu Info..., dragué depuis la palette des menus. De même, le panneau standard d'impression a été inclus en connectant le menu Print... à la fenêtre TVA en invoquant la méthode déjà existante PrintPSCode: (*voir photo ci-dessus*). Toute l'opération s'apparente à un jeu de construction donc chaque étape a pu être testée.

Génération du code

L'IB génère un fichier .nib qui contient les informations relatives à l'interface et aux connexions interobjets. Il faut alors se positionner sur la classe MonObjetTva, puis, à l'aide de la commande Unparse du menu Operation, l'Interface Builder va engendrer le fichier interface (MonObjetTva.h) et le squelette du fichier des méthodes (MonObjetTva.m).

En voici les contenus respectifs :

```
Fichier MonObjetTva.h
/* Generated by Interface Builder */
#import <objc/Object.h>
@interface MonObjetTva:Objet
{
 id  prixTTC;
 id  prixHT;
  id  TVA;
  }
   - plus5 : sender;
   - plus18 : sender;
   - calcul : sender;
   - moins18 : sender;
   - moins5 : sender;
@end

Fichier MonObjetTva.m

/* Generated by Interface Builder

*/   #import "MonObjetTva.h"

@implementation MonObjetTva

   - plus5 : sender
  {
return self;  }
   - plus18
  {
return self;
  }
   - plus18 : sender
  {
return self;
  }
```

```
   - calcul : sender
  {
return self;
  }
```

Production du code manquant :

```
#import "MonObjetTva.h"
#import <appkit/appkit.h>
@implementation MonObjetTva
float taux = 1.186; /* déclaration et
                       initialisation de
                       taux à 18,6% */
 - plus5 : sender  /* méthode d'instance */
  {
taux = 1.055;      /* affectation du taux */
return self;
  }    - plus18 : sender
  {
taux = 1.186;
return self;
  }
   -calcul : sender
  {
float hors_taxe;
hors_taxe = [prixHT floatValue];
/* récupération prix hors taxe */
   [TVA setFloatValue:hors_taxe*taux-hors_taxe];
/* Affichage tva */
     [prixTTC setFloatValue: hors_taxe*taux-
hors_taxe];
/* Affichage prix TTC */
return self;
  }
   - moins18 : sender
  {
taux = .824;    return self;  }
   - moins5 : sender
  {    taux = .945
return self;  }
@end
```

Chaque fois que l'on commence une application, l'Interface Builder crée un fichier «nom-de-l'application_main.m» contenant le programme principal de celle-ci. Ce dernier engendre une instance de l'application et assigne un numéro (id) à la variable NXApp. Il charge le fichier .nib de l'application qui reste active tant que le message Terminate: associé par défaut au menu Quit de l'application n'est pas envoyé. A la réception de ce dernier, il libère l'espace mémoire allouée et retourne 0 à l'environnement pour l'informer de la mort du process.

```
# import <stdlib.h>
# import <appKit/Application.h>
void main ()
{
NXApp = [Application New];
 /* noupplication */
[NXApp loadNibSection:"TVA.nib" owner:NXApp];
[NXApp run];
[NXApp free];
exit(0);
}
```

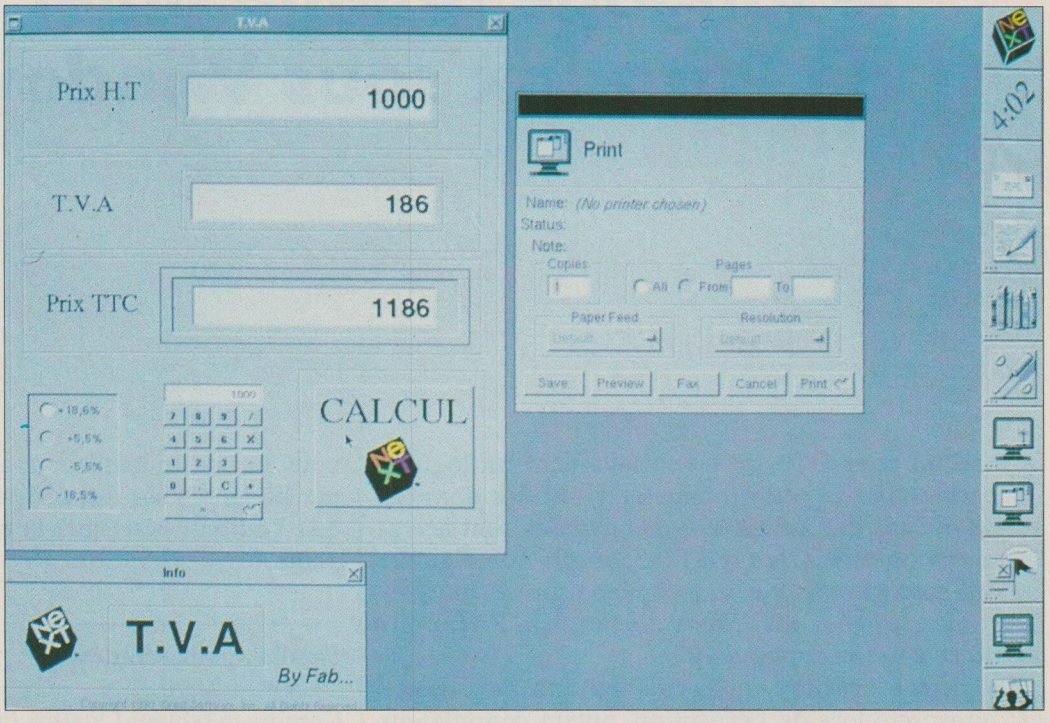
Génération du code exécutable

La commande Make de l'Interface Builder lance la compilation après qu'elle ait créé le fichier Makefile. Si celle-ci s'est bien passée, l'exécutable se trouve généré. Il se trouve alors dans le répertoire du projet avec l'extension .debug. Dans le cas contraire, il faudra déboguer.

L'Interface Builder associe une icône par défaut à

Service lecteurs n° 13 (dernière page)

Application TVA terminée



l'application. Pour la remplacer par une autre personnalisée, il suffit de rouvrir le fichier .nib dans le menu Projet, de choisir l'option Attributs au lieu de File. Quatre cases apparaissent alors, la première contient l'icône par défaut.

Il reste alors à ouvrir la mallette des icônes, et de coller la nôtre à la place. L'application est terminée : il aura fallu en tout et pour tout à peine une heure pour réaliser cette petite application (*voir photo*).

Conclusion

Un ordinateur des années 90 : cette formule résume bien ce qu'est NeXT. Une machine pensée pour répondre aux exigences, voire au rêve de l'utilisateur et de l'informaticien d'aujourd'hui.

Sous Unix, dont l'emploi est transparent, il intègre une technologie de pointe, offrant des possibilités de connexion à des réseaux à haut débit d'informations. Il permet également le traitement numérique de tous les types de données (images, sons, etc.). Les outils multimédias qu'il propose ne relèvent ni du luxe ni du gadget : ce sont de véritables outils de travail quotidien.

L'impression et l'affichage parlent le même langage, PostScript, sans interpréteur. La couche logicielle, notamment NeXTStep, constitue un chef-d'œuvre en matière d'interface graphique, autant par le design que par ses étonnantes fonctionnalités.

NeXT a évolué vers la programmation graphique : l'informaticien de demain sera aussi un «designer». Grâce au concept objet, à cette philosophie graphique de programmation, les temps de dévelop- pement se voient considérablement réduits, ainsi que la durée de maintenance. Dans le même registre, l'atout multitâche d'Unix, interfacé par NeXTStep, permet de suivre en temps réel la pensée du développeur.

Les prix sont certes élevés, mais quelle configuration, quel confort d'utilisation, quel rendement et quelle puissance! NeXT n'est pas un système à «Rustines». Il offre, en version de base, aussi bien dans ses aspects logiciel que matériel, la performance, la qualité, le multimédia et l'ouverture sur l'extérieur. En résumé, essayer NeXT, c'est l'adopter. ❐

	Station	Station Turbo	Station Coul. Turbo	Cube	Cube Server
Moniteur	"17"" mono."	"17"" mono."	"17"" / 21"" Coul."	"17"" mono."	"17"" mono."
Unité Centrale	NeXTStation (25 MHz)	Station Turbo (33 MHz)	Station Coul. Turbo (33 MHz)	Cube (25 MHz)	Cube (25 MHz)
Mémoire Vive	8 Mo	8 Mo / 16 Mo	16 Mo / 32 Mo	16 Mo / 32 Mo	32 Mo / 64 Mo
Lecteur de disq.	2,88 Mo	2,88 Mo	2,88 Mo	2,88 Mo	2,88 Mo
Disque Dur	105 Mo	250 Mo / 400 Mo	250 Mo / 400 Mo	400 Mo / 1,4 Go	1,4 Go / 2 x 1,4 Go
Prix HT	37 600 F	42 300 F / 53 700 F	61 700 F / 87 900 F	75 200 F / 104 700 F	104 700 F / 162 400 F

LANGAGE

OBJECTPAL : le langage orienté objet de Paradox sous Windows

ARNAUD DE LA POMMERAYE

Paradox pour DOS est essentiellement un logiciel permettant de gérer interactivement des données stockées dans des tables. Les commandes sont accessibles via un menu de type 1-2-3, tandis que d'autres actions sont pilotées par une combinaison de touches. Les développeurs disposent du langage PAL pour l'écriture de leurs applications.

Ce langage structuré - qui a déjà sept ans d'existence - est interprété lors de l'exécution, aucun compilateur n'étant à ce jour disponible. Avec la venue prochaine de Paradox sous Windows, des modifications ont été apportées à ce langage qui a été rebaptisé ObjectPAL. Que signifie le terme «objet» dans ce nouveau langage? Quels sont ses points communs et ses dissemblances avec le PAL? Peut-on le comparer au Visual Basic?

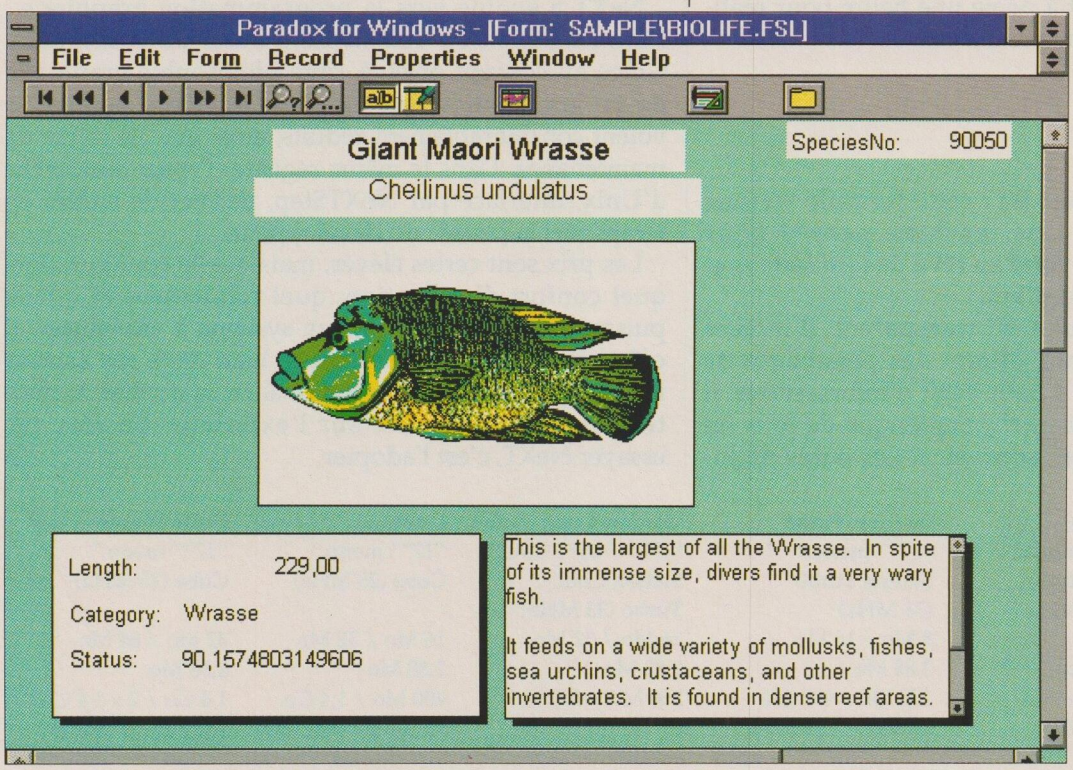
En apportant une réponse exhaustive à ces questions, nous aurons un meilleur aperçu de l'orientation donnée par Borland à ses futurs développements et de l'avenir que l'éditeur réserve à la communauté des développeurs.

Paradox pour Windows

Comme le tableur Quattro Pro Windows, Paradox pour Windows s'appuie sur une interface utilisateur orientée objet (Ooui, Object Oriented User Interface).

Elle possède une barre d'icônes donnant accès aux principales fonctionnalités du logiciel : création de tables et de formulaires, module d'extraction, éditeur de scripts... Tables et formes s'affichent dans des fenêtres, et les enregistrements sont manipulés à l'aide de boutons «magnétoscopes».

On retrouve l'Inspecteur des propriétés de Quattro Pro (DOS et Windows) - assimilable au clic droit de la souris - qui affiche le menu personnalisé des options disponibles pour l'élément sélectionné à l'écran. Paradox pour Windows offre un grand nombre de facilités dans les manipulations «délicates»

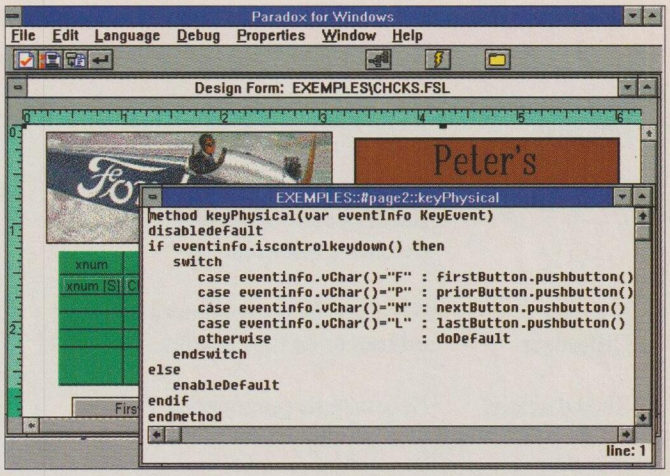


Paradox Windows permet de développer des applications très sophistiquées exploitant pleinement les possibilités graphiques de Windows 3.1.

La famille des SGBD sous Windows va bientôt ouvrir ses portes sur un nouvel outil de développement : Paradox pour Windows. Cette application, que nous avons reçue sous sa version bêta, connaît de nombreuses mutations, notamment au niveau de son langage, PAL (Paradox Application Language). Ce dernier devient objet, ce qui permet d'exploiter au mieux les apports de la version Windows du SGBD de Borland, mais au détriment de la compatibilité avec le PAL sous DOS.

comme la jointure multi table, la génération automatique de rapports et la réalisation d'un publipostage. En revanche, il ne possède pas de générateur d'applications où l'utilisateur serait guidé dans l'élaboration de ses menus (comme dBase IV).

Clé de voûte de l'architecture Boca (Borland Object Component Architecture), Interbase Local Engine est un moteur d'accès aux données qui permet aux applications Borland (dont Paradox Windows) de gérer des données d'origine variée : dBase, Paradox, Btrieve de Novell, tableurs, serveurs de bases de données SQL...Sous Paradox DOS, pour créer un rapport, l'utilisateur devait préalablement élaborer une table «Réponse» avant de concevoir le rapport. Cette étape est supprimée avec la version Windows.



Chaque objet peut recevoir une ou plusieurs méthodes entrées dans une fenêtre spécifique. C'est le cas ici du code ObjectPAL correspondant au formulaire de saisie, régissant les différentes actions des boutons (First, Prior, Next, Last).

L'objet à toutes les sauces

Pour développer une application, le programmeur sous Paradox pour Windows devra entrer dans les arcanes d'ObjectPAL, le langage de programmation orienté objet. Le choix par Borland de l'orientation objet est guidé par deux constatations : ce type de langage simplifie la programmation et autorise la création d'applications hautement sophistiquées. Notons, au passage, que Paradox pour Windows a été entièrement développé en Borland C++, langage lui-même orienté objet (ce qui sera le cas de tous les nouveaux produits Borland).

Paradox pour Windows possède deux types d'objets : les objets de base (script, table, formes, graphiques, rapports et QBE), également appelés «UFO» (User Familiar Objects) et les objets secondaires (champ d'une table, image, titre, grille, bouton, etc.). D'une manière générale, tout ce qui est affiché à l'écran constitue un objet. Chaque objet possède des propriétés (emplacement, titre, couleur, format, etc.) et des méthodes. Celles-ci définissent les actions qui seront déclenchées lorsqu'un événement approprié survient. Propriétés et méthodes se trouvent regroupées dans une même entité nommée «encapsulation». Elles sont stockées dans une base de données attachée au formulaire de l'application.

Comme pour les objets, Paradox Windows distingue deux types de méthodes. D'une part, les méthodes «préprogrammées» qui s'associent de facto à l'objet (plusieurs centaines en tout). Par exemple, le bouton qui s'enfonce lorsque l'on clique dessus. Ces différentes actions, exécutées par défaut (instruction DoDefault), ne nécessitent aucune programmation pour le développeur qui peut les inhiber (instruction «DisableDefault»). D'autre part, les méthodes écrites par le programmeur en ObjectPAL. C'est la programmation même de l'application. Nous verrons ultérieurement comment.

Pour alléger le code, Paradox Windows inclut des ressources Windows comme les boîtes de dialogue, les boutons radio, les cases à cocher, les barres de défilement, etc. Là encore, le développeur n'a pas à s'en préoccuper, ces différents constituants de l'interface utilisateur étant automatiquement activés en fonction de l'événement donné. Il suffit de changer une option dans

la méthode rattachée à l'objet pour que la ressource Windows associée soit également (et automatiquement) modifiée. Cette mise à jour en tâche de fond représente assurément l'un de ses points forts.

L'Object Tool

Pour faciliter la gestion des différents objets d'une application, Paradox Windows possède l'Object Tool. Celui-ci affiche dans une fenêtre les liens entre les différents objets sous forme d'arborescence afin de bien faire apparaître les appartenances. L'attribut souligné indique que l'objet possède une méthode programmée par l'utilisateur, ce dernier pouvant la modifier par simple appel de l'Inspecteur des propriétés (clic droit). Autre outil bien pratique, la génération automatique d'un rapport présentant toutes les méthodes définies dans une forme.

A tout moment, le développeur peut passer du mode édition au mode exécution en cliquant sur l'icône DoIt symbolisée par un éclair. Paradox Windows «compile» l'application et produit un fichier .DLL interne au SGBD. Ce fichier peut être assimilé à du PCode (code précompilé exécuté en interne ou avec un run-time). Comme tout langage de programmation évolué digne

DIFFÉRENCES ENTRE PAL ET OBJECTPAL

Domaine	PAL	ObjectPAL
Langage	Procédural. Les commandes sont stockées dans un script et exécutées linéairement.	Orienté objet. Les méthodes sont attachées à des objets et sont exécutées en réponse à des événements.
Interface Utilisateur	Nécessité de programmer l'interface utilisateur de l'application.	Outils de dessin permettant de placer les objets de l'interface utilisateur.
Evénementiel	Nécessité de programmer l'attente d'un événement (WAIT).	Gestion interne des événements.
Entrée-sortie	Affichage du texte à l'écran @10,15 ?? "Bonjour !"	Affichage du texte dans un objet Montext.text="Bonjour !"
	Nécessité d'écrire du PAL pour piloter l'interface utilisateur : émulation des touches de fonctions et autres touches du clavier (F1, DELETE...)	Utilisation de fonctions : Ecran1.Action.MoveScrollPageDown
Variables	Type «dynamiques». Variables système autolib, retval...	Type «statiques». Aucune variable système.
Gestion mémoire	Nécessité d'écrire du PAL pour gérer la mémoire lorsqu'une application s'exécute.	La mémoire est gérée par Windows.
Procédures	Nécessaires pour sauvegarder de la mémoire et pour améliorer la vitesse 'exécution.	Ne sont d'aucune utilité pour l'optimisation.
Tables	Les tables sont affichées sur l'espace de travail avant de pouvoir les exploiter.	L'utilisateur peut travailler sur des tables sans les afficher à l'écran.
Gestion erreurs	Ecriture de la procédure errorproc.	Utilisation de la boucle TRY...FAIL.

de ce nom, ObjectPAL possède un débogueur permettant très simplement d'indiquer visuellement des points d'arrêt et de consulter - lors de l'exécution - l'état des différentes variables.

En termes de performance, il paraît assez difficile de juger un langage en cours de développement (version bêta 3.51 B). Toutefois, notre impression fut excellente tant ObjectPAL semble «obéir» parfaitement et dans les plus brefs délais aux événements qui surviennent. En revanche, Paradox Windows semble réclamer beaucoup de mémoire pour fonctionner correctement.

De profondes différences avec le PAL

Bien que conçus par le même éditeur, PAL et ObjectPAL constituent deux langages radicalement opposés du fait même qu'ils appartiennent à deux produits évoluant dans deux mondes différents. Le premier sous DOS et le second sous Windows. Au contraire de PAL, ObjectPAL est donc approprié à Windows.

La manière de concevoir une application se révèle totalement différente. Avec Paradox DOS, le script est enregistré dans un fichier ASCII (autant de scripts qu'il y a de procédures). Sous Paradox Windows, le formulaire de saisie et les objets qui le composent constituent le coeur du processus de définition. Le code ObjectPAL des différentes méthodes (ou actions) associées à un objet est entré dans une fenêtre.

Chaque méthode se trouve assimilée à une procédure, enregistrée dans un format interne qui n'est pas l'ASCII. Voici un exemple :

```
method pushButton(var eventInfo Event)
action(datapriorRecord)
endmethod
```

Rattachée au bouton Précédent, cette méthode déclenche l'affichage de l'enregistrement précédent.

Un script ObjectPAL ne contient aucune référence à l'interface utilisateur. Notons que l'on peut également d'écrire (comme avec PAL) un ou plusieurs scripts avec l'éditeur intégré. Ils servent à piloter l'application.

Il suffit de consulter le code source d'une application PAL et d'une autre en ObjectPAL pour constater une

Protocole SPEEDLOAD

3615 FILE

C'est aussi le téléchargement pour les programmeurs

-Des dizaines de programmes sources (Pascal, Assembleur, C, Dbase, Basic)

-Des informations, des documentations techniques

-Et des centaines d'utilitaires à télécharger

3615 FILE est également accessible par 3616 ADLINK*FILE et par abonnement, sur 3624 0166.

Pour recevoir le kit de téléchargement SpeedLoad, écrivez à: ADLINK, 11 Bis rue du Chalet, 92600 ASNIERES. (ou sur 3615 FILE)

■ Je désire recevoir le kit (Joindre 5 F en Timbres)

■ je désire recevoir le Câble+SpeedLoad (75 FTTC)

Nom:
Adresse:

CP: Ville:

Service lecteurs nº 14 (dernière page)

différence d'ordre syntaxique. Toutes les instructions qui interfèrent sur un objet respectent une syntaxe bien particulière:

```
Objet
Global.Objet.Propriétés=valeur ou  fonction
```

Cette syntaxe fait apparaître une caractéristique majeure d'ObjectPAL, la notion d'inclusion. Plus clairement, il s'agit de l'appartenance d'un objet à un groupe. Chaque objet est «autonome» : toutes les actions et toutes les modifications qu'il subit ne s'appliquent qu'à lui seul. Toutefois, comme dans un logiciel de dessin où les objets peuvent être regroupés ou dissociés, Paradox Windows permet d'activer ou de désactiver l'imbrication d'objets.

L'objet «embedded» (imbriqué) possède les mêmes propriétés que celui qui l'inclut. Dissocié, l'objet possède ses propriétés propres. En plus de son orientation «objet», ObjectPAL constitue essentiellement un langage événementiel, Windows détectant l'action de l'utilisation (appui sur une touche, déplacement de la souris) avant même qu'une opération ne soit entreprise par ObjectPAL pour l'objet sélectionné. En PAL 3.5, une application qui commence son exécution initialise un certain nombre de variables et appelle un menu. L'utilisateur sélectionne une option à laquelle sont rattachés un sous-menu ou l'exécution d'une tâche.

C'est l'exemple type d'un «événement à un instant t». Du fait même du caractère multitâche de Windows, une application Paradox Windows peut travailler simultanément sur plusieurs événements : consulter une table, extraire des enregistrements et imprimer un rapport... Une structure qui n'a rien de «modulaire» comme le PAL.

Conséquence immédiate : le code d'une application PAL ne peut être compatible avec l'exécution sous Windows.Un autre atout majeur d'ObjectPAL par rapport au PAL consiste dans l'utilisation des Tcursors. Utilisés comme pointeurs des tables ouvertes, ils facilitent la gestion et la maintenance des tables Paradox dans une application qui n'a pas besoin de les afficher à l'écran, comme sous DOS, pour exploiter les données qu'elles contiennent.

QUELQUES QUESTIONS

☛ **Peut-on exécuter des applications PAL à partir de Paradox Windows?**
Oui, à la condition que Paradox (versions 3.5 ou 4.0) ou son run-time soient présents sur le disque dur. L'utilisateur peut placer les différentes applications dans un dossier, chacune d'elles étant représentée par une icône. Les applications PAL s'exécutent alors dans une fenêtre DOS en mode caractères.

☛ **Peut-on récupérer du code PAL pour le transformer en ObjectPAL?**
Non. Et c'est sans doute la plus grande contrainte que vont rencontrer les développeurs PAL. Ce qui n'est pas le cas de ceux qui passent de QuickBasic à Visual Basic ou de dBase IV 1.5 au compilateur dBase pour Windows.

☛ **Peut-on inclure du code PAL dans une application ObjectPAL?**
Non.

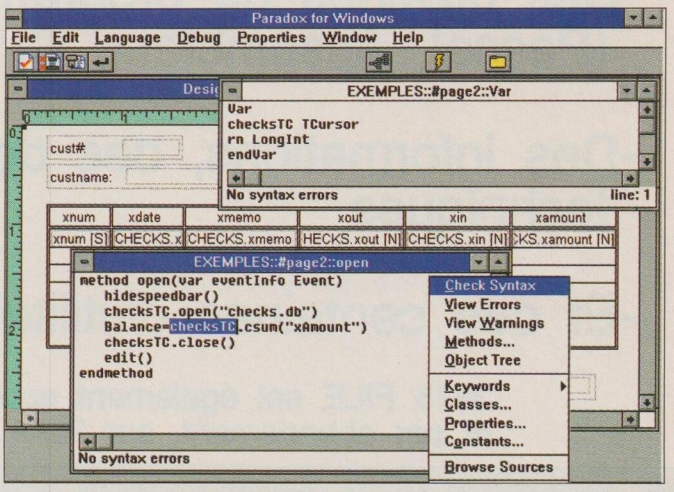
☛ **Les formats de fichier de Paradox Windows sont-ils compatibles avec ceux de la version DOS?**
Oui, mais uniquement avec la dernière version de Paradox, la 4.0. Ce nouveau produit comporte 85 nouvelles commandes et gère les données 10 fois plus rapidement que la version 3.5.

☛ **Existe-t-il un convertisseur PAL-ObjectPAL?**
Non. Et il n'y en a pas de prévu tant les deux langages différèrent. Même réponse à ceux qui souhaiteraient un convertisseur PAL-dBase...

☛ **PAL est-il un langage compilé?**
Non. Une application PAL est précompilée, Paradox exécutant du PCode.

☛ **ObjectPAL est-il un langage compilé?**
Non. Lorsque l'on «compile» une application, Paradox crée en réalité une librairie de fonctions (DLL) dans un format interne. Donc, pas de «.EXE». Borland laisse sous-entendre que les prochaines versions de Paradox Windows pourront créer de véritables DLL. Toujours est-il qu'une application ObjectPAL nécessite la présence d'un run-time pour fonctionner. Le développeur devra se le procurer (une fois pour toutes) et aura la possibilité d'effectuer autant de diffusions qu'il le souhaite sans verser de royalties.

☛ **Y aura-t-il un compilateur ObjectPAL?**
Pas de réponse affirmative de la part de Borland qui, pourtant, laisse percevoir qu'une équipe travaillerait sur le projet.



Il est possible de demander à Paradox Windows de vérifier la syntaxe des différentes instructions d'une fenêtre, fonction très pratique pour les nouveaux venus à ObjectPAL...

Réussir vos choix de progiciels

- ***L'information rapidement accessible avec***

- la banque de données vidéotex consultable par minitel
  par le **36 29 00 16** (accès direct) - 12 000 progiciels recensés
- les catalogues de progiciels
  . Gestion de données (376 progiciels) - Réf H9105 - Prix : 550 FF TTC
  . Génie logiciel, outils de développement d'applications (369 progiciels)
  Réf H9104 - Prix : 600 FF TTC
  . Outils de développement et gestion des données sous UNIX (212 progiciels)
  Réf U9201 - Prix : 630 FF TTC
  . CASE and development tools (149 progiciels) - Réf E9104 - Prix : 780 FF TTC

- ***L'information comparative et prospective***

- des études comparatives
  . Les SGBD orientés objet
  . Les environnements de développement orientés objet
  . Les serveurs SQL en architecture client/serveur
  . Les SGBD répartis
  . Les environnements de développement sous UNIX
- des dossiers prospectifs
  . Vers l'industrie du logiciel
  . Les architectures coopératives

- ***Le conseil en progiciel***

- une démarche originale de conseil, appréciée par sa simplicité d'appplication, son orientation pratique et axée sur l'offre du marché. Les centres de compétence du CXP couvrent tous les domaines de gestion, le génie logiciel et les environnements de développement.

Service lecteurs n° 15 (dernière page)

CXP
La compétence en progiciel

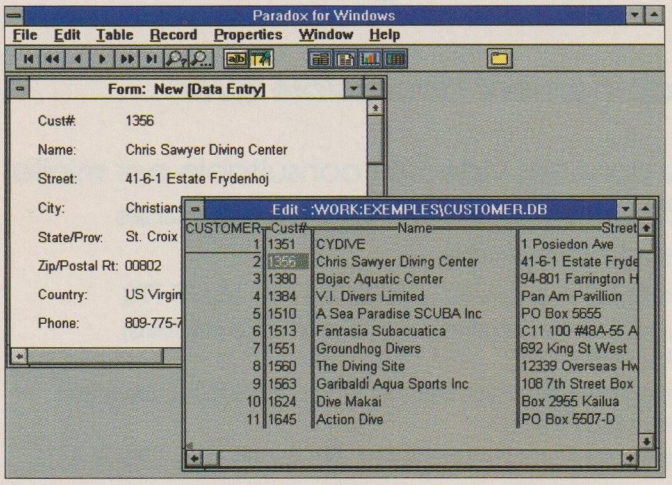
LS

Vous désirez recevoir une documentation sur :

- ❑ la banque de données et les catalogues de progiciels
- ❑ les études comparatives et les dossiers prospectifs
- ❑ le conseil

M ..... Fonction .....
Société ....
Secteur d'activité ....
Adresse ....
..... Tél .....

Pour en savoir plus, retournez ce coupon au CXP - J.L. Alarçon - Direction Commerciale - 19-21, rue du Rocher - 75008 Paris ou appelez au (1) 43 87 90 28



Comme la version DOS, Paradox Windows affiche les tables dans des fenêtres et offre un module très puissant d'extraction de données (QBE).

L'accès aux enregistrements s'avère immédiat et s'effectue de façon transparente pour l'utilisateur final. Ci-dessous un exemple de code ObjectPAL exploitant un Tcursor :

```
var
        tc Tcursor
        client String
        i SmallInt
endVar
tc.open("clients.db")
for i from 1 to 100
        tc.fieldValue("Nom",client)
        if client="Dupont" then
                quitloop
        else
                tc.nextRecord()
        endif
endFor
```

PAL constitue un langage interprété, alors qu'ObjectPAL correspond à un langage «compilé» (voir l'enacdré sur les questions-réponses).

PAL est un langage basé sur l'utilisation de commandes pilotant l'application. ObjectPAL se différencie de ce modèle, les commandes qu'il possède ne servant qu'à définir l'architecture de fonctionnement des objets. Telles les structures IF...THEN...ELSE, SWITCH...CASE, WHILE...FOR, etc.

Il y a bien d'autres distinctions entre les deux langages. En voici quelques-unes :

• PAL emploie des variables «dynamiques», alors qu'ObjectPAL gère des variables «statiques» (comme le C ou le Pascal).

• PAL comporte trois variables réservées (retval, autolib et errorproc), alors qu'ObjectPAL n'en comporte aucune.

• On peut se dispenser de déclarer des variables en PAL alors qu'il est fort recommandé de le faire avec ObjectPAL. Le type de variables conditionne le comportement de la méthode et permet d'optimiser son exécution.

• PAL utilise des variables de type «global» et «private», alors qu'ObjectPAL ne sait gérer que celles rattachées à des objets. Elles existent uniquement durant l'activation d'un objet.

La liste est longue. Pourtant, PAL et ObjectPAL ont des points communs.

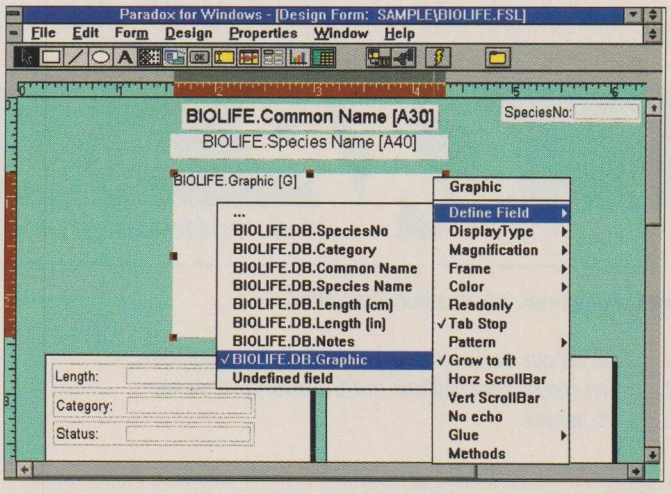
Quelques similitudes

Les développeurs en PAL trouveront dans ObjectPAL quelques ressemblances avec leur langage. Ainsi, pour afficher un message, la syntaxe employée n'est pas radicalement différente : MESSAGE "Bonjour !" (pour PAL) et MESSAGE("Bonjour !") (pour ObjectPAL). La commande se transforme en fonction. On retrouve bien d'autres exemples de «migration» bâtis sur ce modèle (comme SLEEP x).

Certaines commandes et fonctions (une trentaine en tout) sont même compatibles entre les deux langages (voir encadré). On y retrouve notamment les fonctions de gestion de tables NRECORDS(), CSUM() ou ISEMPTY()...

De plus, les structures de programmation, qui constituent le coeur du code source comme IF-THEN-ELSE-ENDIF, n'ont pas changé.

Tout langage a ses qualités et ses défauts. La question est de savoir comment va s'effectuer la migration du PAL vers l'ObjectPAL.

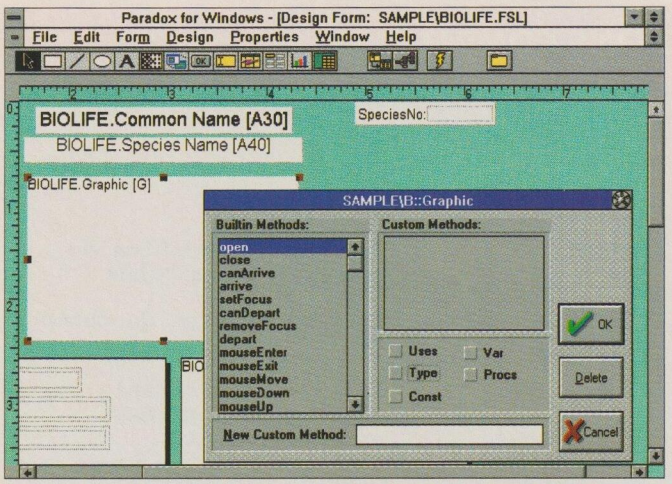


L'Inspecteur des propriétés permet également de sélectionner un champ d'une table pour le placer dans un formulaire de saisie.

COMMANDES ET FONCTIONS COMPATIBLES

Fonction	Objet
ADD	Ajoute des données dans une table
cAverage	Retourne la moyenne d'une série de valeurs d'un champ d'une table
cCount	Retourne le nombre total de valeurs non vides d'une série de valeurs d'un champ d'une table
cMax	Retourne le maximun d'une série de valeurs d'un champ d'une table
cMin	Retourne le minimum d'une série de valeurs d'un champ d'une table
cNPV	Retourne la valeur actuelle d'un bien en fonction d'un taux d'intérêt
COPY	Copie une table
cSamStd	Retourne l'écart simple d'une série de valeurs d'un champ d'une table
cSamVar	Retourne la variance simple d'une série de valeurs d'un champ d'une table
cStd	Retourne l'écart type d'une série de valeurs d'un champ d'une table
cSum	Retourne la somme d'une série de valeurs d'un champ d'une table
cVar	Retourne la variance d'une série de valeurs d'un champ d'une table
DELETE	Supprime une table
EMPTY	Détruit tous les enregistrements d'une table
FieldName	Retourne le nom du champ d'une table en fonction de son numéro
FieldNo	Indique la position du champ dans la table
FieldType	Précise le type du champ de la table
IsEmpty	Indique si une table contient ou non des enregistrements
IsEncrypted	Indique si une table est cryptée ou non
IsShared	Précise si une table peut être ouverte en mode partagé
IsTable	Indique si une table existe dans la base de données
nFields	Retourne le nombre de champs de la table
nKeyFields	Retourne le nombre de champs indexés dans la table
nRecords	Indique le nombre total d'enregistrements dans la table
PROTECT	Protège une table par un mot de passe tout en encryptant les données
REINDEX	Recrée un ou plusieurs fichiers d'index
REINDEXALL	Reconstruit tous les fichiers d'index d'une table
RENAME	Renomme une table
SUBSTRACT	Soustrait des enregistrements d'une table vers une autre
TABLERIGHTS	Indique quels sont les droits de l'utilisateur sur une table

Les commandes sont écrites en majuscules.



Chaque élément de l'application est assimilé à un objet possédant une ou plusieurs propriétés et méthodes. C'est grâce à l'Inspecteur des propriétés (clic droit de la souris) que le développeur les définie rapidement.

Malgré les différences prononcées entre ces deux derniers, il est probable que les développeurs professionnels auront vite fait de s'adapter à ce nouveau langage. Mais ils doivent réécrire les applications conçues pour Paradox DOS, ce qui peut représenter plusieurs mois-homme de perdus. Une solution intermédiaire consiste à lancer les applications PAL sous une fenêtre DOS. Leur plus grande difficulté résidera dans l'adaptation d'une nouvelle méthode de saisie.

Abandonner les combinaisons de touches et les différents mécanismes acquis avec le temps n'est pas chose aisée!

Conscient de cet état de fait, Borland préfère porter l'accent sur la richesse des possibilités offertes par ObjectPAL plutôt que de reconnaître et de se lamenter sur la non-récupération du code existant et de la migration d'applications DOS vers Windows. Le message sera difficile à faire passer. D'autant que, si l'on compare ObjectPAL à d'autres langages en pleine évolution comme le basic ou dBase (tous deux orientés objet et fonctionnant sous Windows), ObjectPAL fait triste mine.

Des points communs avec Visual Basic

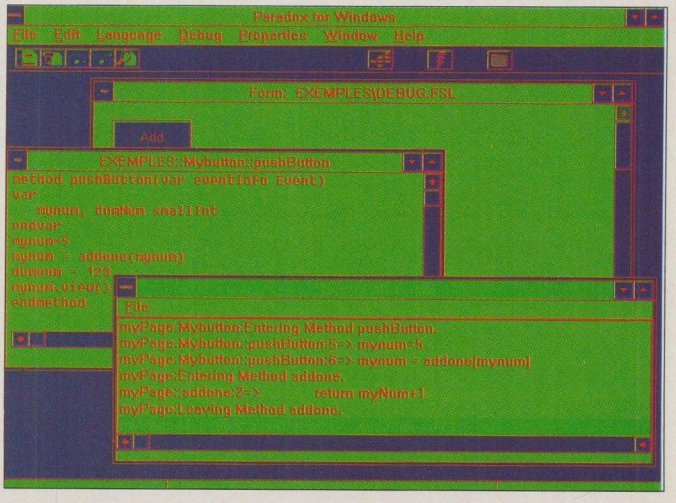
On peut comparer ObjectPAL et Visual Basic sur deux points : l'ergonomie et le langage. Paradox Windows comme Visual Basic ont une ergonomie de fonctionnement très similaire. D'une part, au niveau des moyens techniques offerts aux développeurs, regroupés dans une boîte à outils et permettant de créer la plupart des objets propres à l'environnement Windows (dessins, textes, tableaux, boîtes de dialogue, boutons, images, etc.). D'autre part, au niveau de l'entrée du code propre à l'objet.

Les deux produits affichent une fenêtre de saisie dans laquelle le développeur compose son programme au sein d'une procédure dont les lignes d'entrée et de sortie se trouvent déjà affichées. A tout moment, le développeur peut tester son application en effectuant des aller et retour entre module de conception et module d'exécution.

Toutefois, Visual Basic est un langage de programmation alors que Paradox Windows constitue avant tout un SGBD. Ce dernier bénéficie donc d'un noyau de fonctions (une cinquantaine) propres à la gestion de données dans des tables, importante lacune de Visual Basic. Il faudra rajouter au langage de Microsoft une extension (Professional Toolkit) pour retrouver, par exemple, le séquentiel indexé. En revanche, Visual Basic dispose d'un Run-Time qui fait défaut à Paradox.

dBase s'adapte mieux que PAL à l'environnement objet. En effet, - comme nous l'expliquerons en détail dans le prochain numéro - la structure même du langage dBase n'est pas modifiée de sorte que la programmation sous Windows soit aussi simple que sous DOS. Les options propres à la gestion des objets ont été rajoutées et viennent s'insérer naturellement dans une commande dBase. Ce qui n'est pas du tout le cas entre PAL et ObjectPAL.

Ainsi, nous pouvons affirmer que les développeurs en langage dBase sont sur bien des points (add-in, compilateur...), et principalement sur l'environnement graphique objet, plus avantagés que les programmeurs PAL. Le passage du PAL à ObjectPAL comporte obligatoirement une remise en question totale de leur méthode de programmation, avec une perte de l'existant. Comme quoi, générer un nouveau langage de programmation n'est pas toujours sans risque.



ObjectPAL Tracer enregistre dans une fenêtre les différentes étapes de l'exécution pendant la mise au point de l'application.

NOUVELLE VERSION 3.0
Compatible Windows.

EVERLOCK, L'ANTIVOL LOGICIEL

FRUIT DE NOMBREUSES ANNÉES D'EXPÉRIENCE, EVERLOCK S'IMPOSE AUJOURD'HUI COMME LE LOGICIEL DE CHOIX POUR PROTÉGER TOUS VOS PROGRAMMES DOS ET WINDOWS :

- Très haute protection contre le désassemblage, la copie octet par octet, reinstall, copyll PC, copywrite, Option Board, etc... NOUVEAU
- Encryption unique de vos fichiers et de vos programmes. NOUVEAU
- Paramétrage très souple avec compteurs du nombre de copies, d'utilisateurs, de jours...
- Mises à jour à distance de vos logiciels (version professional).
- Transparence totale pour l'utilisateur en mono-poste ou réseau.
- Installation rapide et conviviale sans disquette spéciale.
- Protection Anti-virus intégrée. NOUVEAU
- Très faible coût par copie.

VERSION PROFESSIONAL
3490 F HT
(4139,14 F TTC)
POUR 120 COPIES

Pour commander une disquette de démonstration appelez ou écrivez à :
Tél. (1) 40 99 28 00 - 2, rue des Bourets - 92150 Suresnes - Fax : 40 99 28 88

INNOSOFT

3490 F HT (4139,14 F TTC) pour 120 utilisations. 9950 F HT (11800, 70 F TTC) pour la version illimitée. Documentation en français.

BON DE COMMANDE - A renvoyer à INNOSOFT 2, rue des bourets - 92150 Suresnes
Veuillez trouver ci-joint un chèque de 50 F TTC et me faire parvenir la disquette de démonstration EVERLOCK ainsi que sa documentation.
Nom : .....Société : .....
Adresse : ..... Code postal : |_|_|_|_|_| Ville : .....
PCE 06/92

Service lecteurs n° 16 (dernière page)

RT PATCH

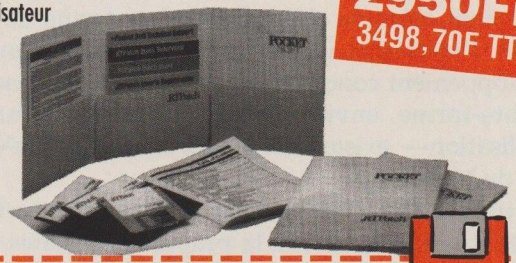
La meilleure solution pour mettre à jour vos logiciels.

RT PATCH vous permet de fabriquer et d'envoyer des "Patch" à vos utilisateurs qui ne contiennent que les différences entre les nouveaux et les anciens fichiers

- Fabrication automatique de "Patch" sur tout type de fichiers
- EXE, bases de données , textes et tableurs
- Diminution de la taille des fichiers envoyés sur disquette ou modem
- Interface utilisateur simple et conviviale.
- Systèmes de vérification intégrés.
- Compatible DOS, OS 2 et Windows

Installation automatique sur le système de l'utilisateur

RT PATCH
est un produit de Pocket Soft
2950F HT
3498,70F TTC

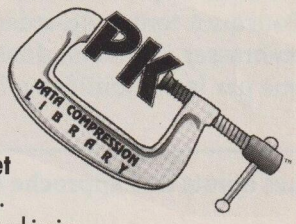


BON DE DOCUMENTATION
☐ *OUI, je désire recevoir gratuitement une documentation complète sur les différents produits de Pocket Soft*
Nom:.....
Société:.....
Adresse:.....
Code Postal:..... Ville:.....

INNOSOFT
A renvoyer à
INNOSOFT
2, rue des Bourets
92150 SURESNES
Tél. 40.99.28.00
Fax. 40.99.28.88
MS 06/92

PK ZIP LIBRARY

Gagnez de 50 à 90% d'espace dans vos applications



Incorporez les dernières technol PK WARE (fabricant de PK Zi , compression dans vos applications et gagnez ainsi de 50 à 90% d'espace.

- Compresse tout type de données, ASCII ou binaire,
- Contrôle des données d'entrée et de sortie des routines de compression depuis l'application appelante,
- Compatible avec la plupart des langages C, C++, Pascal, Assembleur, Basic ou Clipper,
- Utilisable sous Windows et OS 2 en mode réel ou protégé,
- Ajoute 7K à la taille de l'éxécutable,
- Pas de royalties

PK ZIP
est Déposé par PK WARE
2690F HT
3190,34F TTC

BON DE DOCUMENTATION
☐ *OUI, je désire recevoir gratuitement une documentation complète sur les différents produits de PK WARE*
Nom:.....
Société:.....
Adresse:.....
Code Postal:..... Ville:.....

INNOSOFT
A renvoyer à
INNOSOFT
2, rue des Bourets
92150 SURESNES
Tél. 40.99.28.00
Fax. 40.99.28.88
MS 06/92

SYSTÈME

SUR LES TRACES DE NEXTSTEP 3.0

L'outil pour utilisateur et développeur

MAXIME DADDLER

Imaginez que, avec ce que coûterait la fabrication du plus puissant ordinateur du monde, il soit possible de construire un millier d'appareils mille fois moins puissants. En confiant ces appareils à mille personnes créatives, on obtiendrait bien plus que le travail d'une seule personne aux commandes du computer le plus performant. Pourquoi? Parce que les gens sont créatifs! Ils utilisent les outils au-delà de l'imagination des fabricants. Et quand l'un d'eux trouve une nouvelle utilisation de l'outil, il peut en faire profiter les 999 autres.» Il suffit, pour s'en convaincre, de comparer le nombre d'applications disponibles sur Connection Machine et sur Macintosh.

C'est ce raisonnement, résumé par Steve Jobs, fondateur d'Apple puis de NeXT, que l'on retrouve à l'origine même de la naissance des micro-ordinateurs. Il guida les premiers pas du Macintosh et se situe aujourd'hui au cœur même de la conception de NeXT et de NeXTStep. La programmation et le développement orientés objet sont basés sur un principe similaire. Pourquoi tenter de créer un programme cherchant à embrasser l'univers, dont la puissance ne serait égalée que par le monolithisme ?

Les atouts de l'approche orientée objet

Alors que mille applications orientées objet compactes, accomplissant le minimum de fonctions, sont bien plus simples à mettre en œuvre et beaucoup plus efficaces. Pourquoi développer des applications finies et fermées, des environnements repliés sur eux-mêmes ? Alors que des objets simples et précis suffisent amplement...

Chaque utilisateur a des besoins différents, et sait généralement mieux que le programmeur ce dont il a besoin.

En reliant entre eux plusieurs objets, les utilisateurs peuvent créer les fonctions qui répondent le mieux à leurs attentes. En ajoutant plusieurs fonctions les unes aux autres, ils construisent eux-mêmes leur propre environnement, entièrement modulaire et reconfigurable. Les ordinateurs individuels, et l'approche orientée objet, ont permi d'améliorer la production individuelle. Mais, pour compléter l'exemple de Steve Jobs et aboutir à une informatique «interpersonnelle», il est nécessaire qu'objets, applications, environnements, mais aussi utilisateurs et programmeurs puissent communiquer facilement.

Les conditions d'une informatique interpersonnelle

La notion d'Informatique Interpersonnelle comporte trois parties : communication, collaboration et contenu. Pour améliorer la collaboration entre utilisateurs et développeurs ainsi que le contenu — c'est-à-dire la productivité du groupe — il faut effectivement commencer par communiquer!

Les ordinateurs individuels doivent donc obligatoirement être reliés entre eux. Il est nécessaire que les applications intègrent plus de solutions multimédias, que les objets puissent s'envoyer des messages et, ainsi, échanger des informations. De l'objet à l'utilisateur en passant par la machine elle-même, la démonstration du fondateur de NeXT, et l'analyse de nombreux créateurs, pourra alors prendre forme.

Pour mettre en application une telle théorie, développeurs et utilisateurs doivent disposer d'une même plate-forme de travail multitâche, multi-utilisateur, totalement orientée objet. Cette plate-forme devra être munie d'interfaces utilisateur et d'environnements de développement conçus dans ce sens. De tels ensembles —plate-forme, environnement de programmation et d'utilisation— existent déjà : les machines NeXT équipées de NeXTStep!

Actuellement, l'utilisateur dispose d'interfaces et d'outils performants sur la plupart des plates-formes. Le développeur, quant à lui, ne bénéficie pas encore des mêmes attentions de la part des constructeurs. Seul NeXT propose, avec NeXTStep, un ensemble complet construit autour de Mach (de l'université de Carnegie

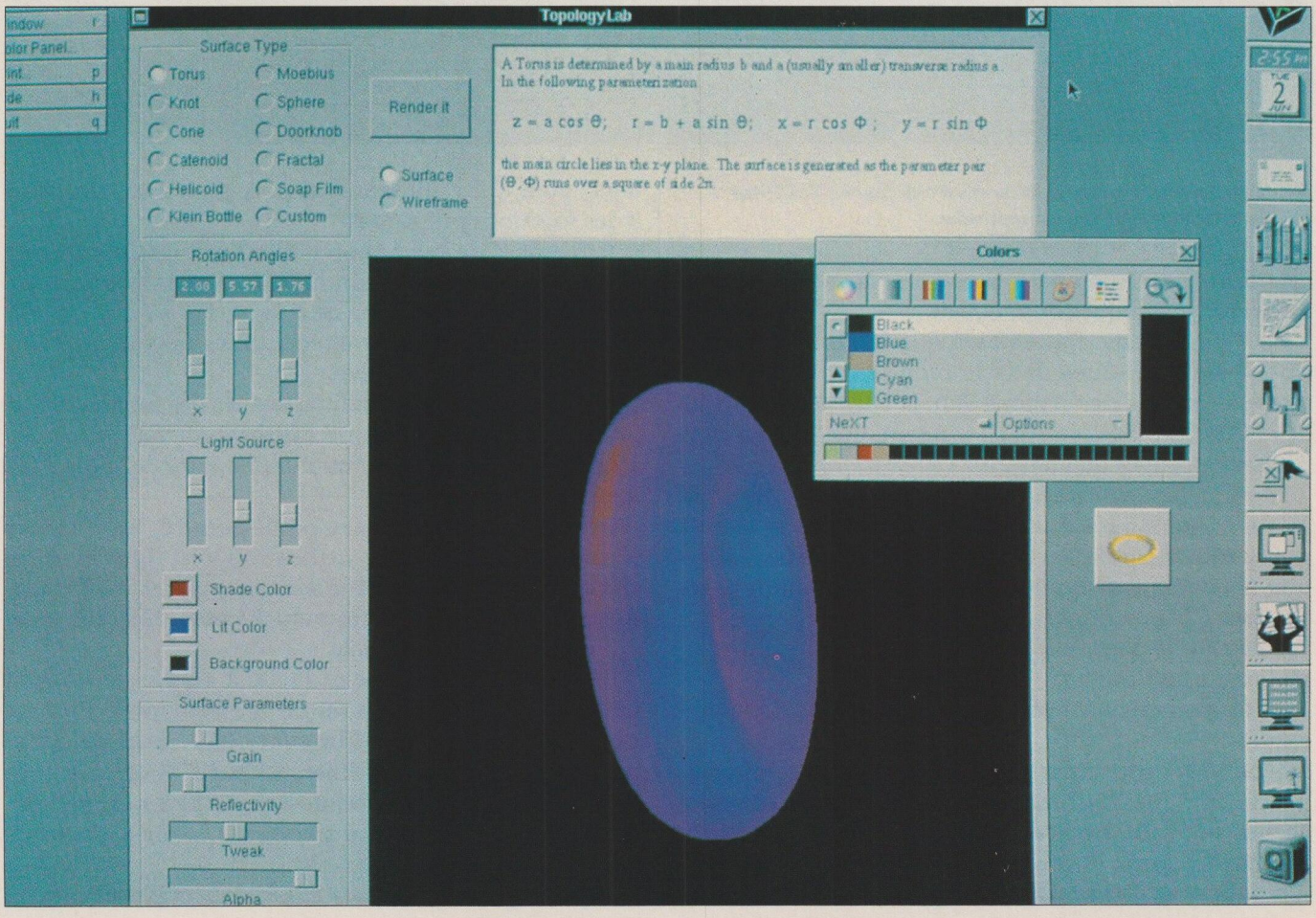
Utiliser Unix sans l'apprendre, programmer sans écrire la moindre ligne de code ou bénéficier d'applications multimédias... Aujourd'hui, un seul environnement répond simultanément aux exigences de l'utilisateur et du développeur. Homogène, convivial et totalement orienté objet, NeXTStep est actuellement l'interface utilisateur graphique et l'environnement de programmation le plus réputé du monde.

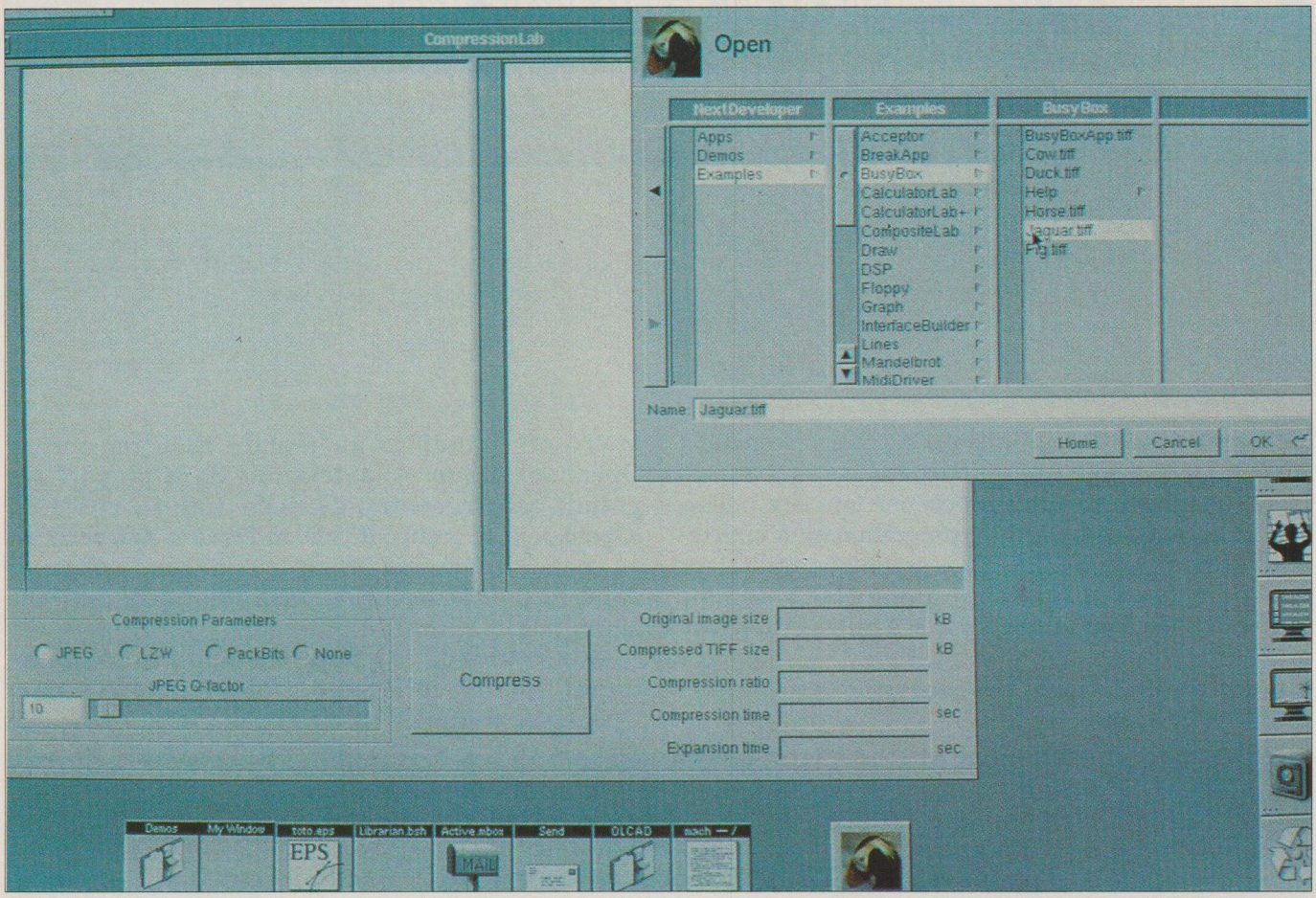
Mellon), logiciel système orienté objet choisi récemment par l'OSF (Open Software Foundation).

Cet ensemble regroupe à la fois une interface utilisateur et un environnement de programmation entièrement orientés objet. L'interface graphique utilisateur des ordinateurs NeXT est propre à NeXT, comme l'interface du Macintosh l'est à Apple, l'interface Microsoft Windows au monde DOS ou encore OpenLook à Sun. A la demande de la plupart des grands constructeurs, de l'OSF (Motif) et d'USL (OpenLook), le groupe de travail 1201.1 d'X/Open s'est vu confier la tâche de définir une interface utilisateur standard et de la déposer à l'IEEE afin qu'elle devienne une norme ISO, Ansi, ...

Les membres de ce groupe ont d'ores et déjà sélectionné quatre interfaces : MS-Windows, Mac, OpenLook et Motif. Ils n'ont pas choisi NeXTStep,

Logiciel graphique. L'une des nombreuses applications standards sous NeXTStep.





Un maximum d'icônes et de menus d'aide font de NeXTStep un environnement facile à apprendre.

pourtant réputé comme l'une, voire la meilleure, des interfaces graphiques du moment.

Pourquoi? Sans doute parce que le nombre de stations NeXT vendues dans le monde est encore loin d'atteindre celui des stations Sun, et encore moins celui des PC ou des Macintosh. Cependant, quantité ne rime pas toujours avec qualité.

Alors, comment caractériser une bonne interface utilisateur? Une interface utilisateur est un moyen simple pour l'utilisateur de demander à son ordinateur d'effectuer les opérations qu'il a à faire. Pour répondre à ses attentes, elle doit donc :

- être facile à retenir, pour réduire les temps de formation de l'utilisateur;
- être pratique à employer afin d'augmenter son efficacité;
- posséder une présentation et une structure homogènes, afin que l'utilisateur ne se sente pas dépaysé à chaque nouvelle application et réduise ainsi son temps d'apprentissage;
- supporter l'ensemble des opérations liées au système, pour que l'utilisateur dispose d'un accès simplifié à des fonctions puissantes;
- disposer de métaphores montrant une similitude avec des actions du monde réel. Ainsi, l'utilisateur n'aura à emmagasiner que le strict minimum de connaissances et pourra effectuer des opérations complexes de manière intuitive;
- disposer enfin de similitudes avec des interfaces utilisateur connues, afin que l'utilisateur habitué à ces autres interfaces n'ait qu'un minimum de données à réapprendre.

Excepté pour le dernier point énoncé, les qualités d'une bonne interface utilisateur ont donc peu de choses à voir avec les standards en vigueur. NeXTStep, lui, correspond parfaitement à cette définition de l'interface graphique idéale.

Il est aussi homogène et facile à employer que les interfaces du Macintosh ou de MS-Windows. Et pour cause : les éléments de l'interface NeXTStep résultent de plusieurs années de recherches sur l'interface utilisateur. L'expérience du Macintosh a d'ailleurs été entièrement prise en compte dans ce projet, pour des raisons historiques évidentes.

Cependant, NeXTStep va aussi beaucoup plus loin en

supportant notamment le multitâche sous Unix et les communications à haut débit. Ses deux principaux atouts demeurent néanmoins sa flexibilité et sa capacité d'évolution, dues à sa programmation orientée objet. A ce sujet, de nombreux progrès ont d'ailleurs été enregistrés depuis la version préliminaire 0.8 présentée en 1988.

Les diverses applications utilisateur, les outils de développement et d'administration système ainsi que les nombreux utilitaires sont dorénavant parfaitement accessibles à travers l'interface utilisateur.

Premier contact avec NeXTStep : au démarrage de la machine, Workspace Manager —point central de NeXTStep pour l'utilisateur— affiche automatiquement l'Application Dock, puis le File Viewer et, enfin, son menu. L'Application Dock se présente sous la forme d'une fenêtre verticale qui s'affiche à la droite de l'écran. Elle contient 13 icônes, mini-fenêtres de 64 x 64 pixels.

Chaque icône, comme chaque élément sous NeXTStep, possède un dessin en relief. Chacune correspond à un programme ou à une application particulière. La première renferme le Workspace Manager et la dernière, le recycleur.

Workspace Manager ou comment naviguer parmi fichiers et répertoires sous NeXTStep.

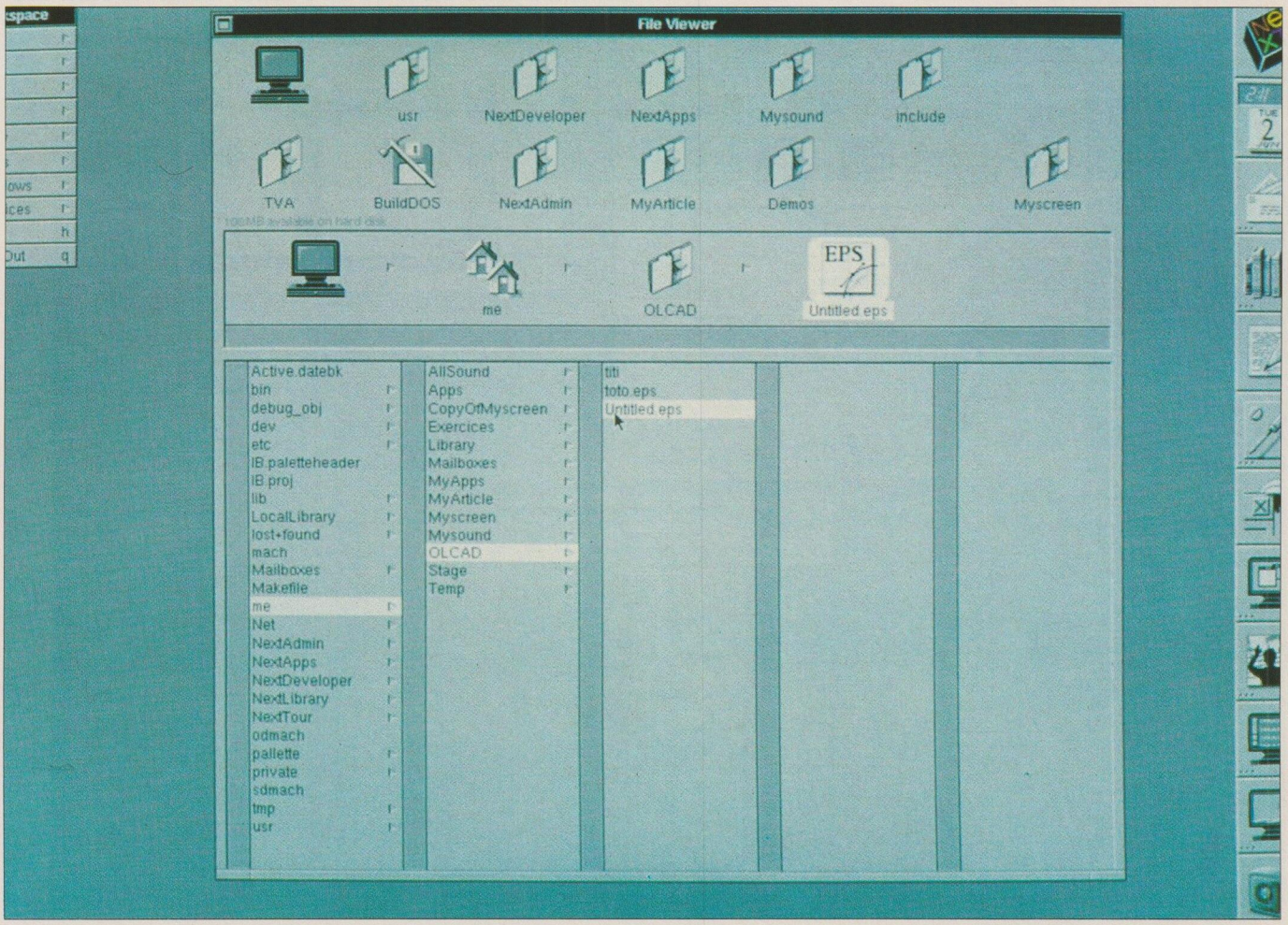
Les vertus du recyclage en informatique

A l'origine, les machines NeXT étaient équipées d'un «trou noir» qui remplissait des fonctions équivalentes à celles de la corbeille du Macintosh. Lorsque l'utilisateur déplaçait une icône près dudit trou noir, celui-ci se mettait à tourner. S'il la lâchait à proximité, elle se retrouvait alors littéralement aspirée.

Aujourd'hui, le trou noir a cédé sa place au recycleur, véritable symbole écologique des années 90. Encore un clin d'œil à l'environnement Macintosh : la poubelle c'est bien, le recycleur c'est mieux !

Les 12 icônes restantes sont réservées à l'utilisateur. Il y place les programmes qu'il désire voir exécutés ou installés automatiquement lors du démarrage de la machine.

Chaque utilisateur possède une configuration qui lui est propre. Son environnement reste protégé par un



mot de passe, ainsi que par des droits d'accès en lecture, écriture et exécution, comme c'est habituellement le cas sous Unix. Sur NeXT, il n'existe qu'une seule sorte de programme : les applications autocliquables ! Leurs icônes sont toutes rangées dans des dossiers et accessibles de la même manière : il suffit de cliquer dessus. Le File Viewer, première fenêtre qui apparaît sur une machine NeXT, sert essentiellement à manipuler les fichiers. Il est divisé en trois parties.

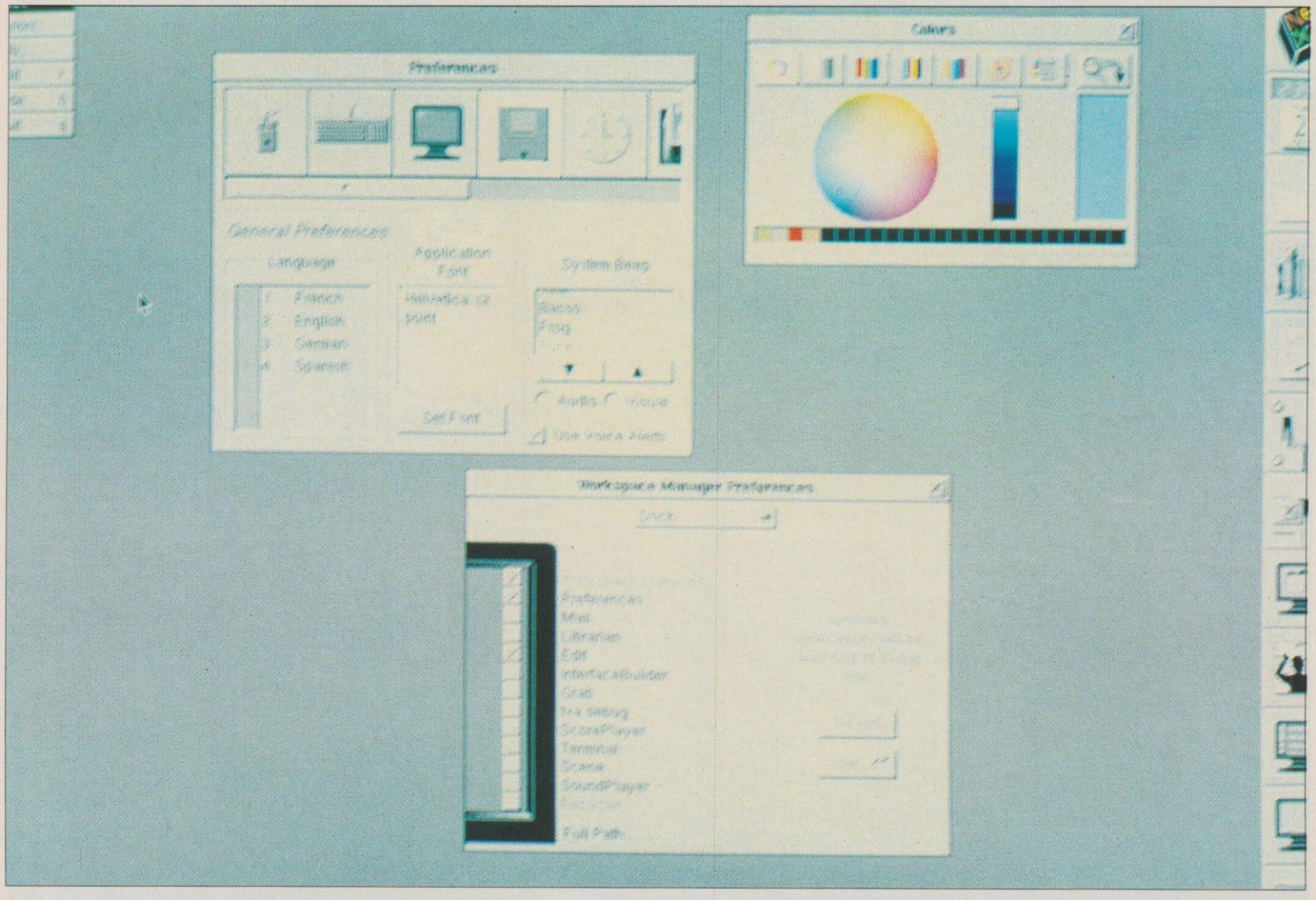
Tout en bas, d'abord, le browser comporte trois modes différents : la vue par icônes, la vue par liste ou le «browser» proprement dit qui permet, lui, de voir simultanément plusieurs niveaux hiérarchiques. Au dessus, l'Icon Path, c'est-à-dire le chemin d'accès des icônes, indique graphiquement où l'utilisateur se trouve dans l'arborescence. Au sommet de la fenêtre, enfin, se situe la Shelf, une sorte d'étagère très pratique sur laquelle l'utilisateur peut poser les icônes des dossiers ou des fichiers auxquels il souhaite faire appel plus facilement ou plus fréquemment. La conception et les principes mêmes de ce système de navigation hiérarchique, parmi les fichiers et les répertoires, reposent sur une étude précise visant à obtenir une amélioration par rapport à l'interface de référence : celle du Macintosh.

NeXTStep : un environnement entièrement reconfigurable par l'utilisateur.

Une comparaison révélatrice

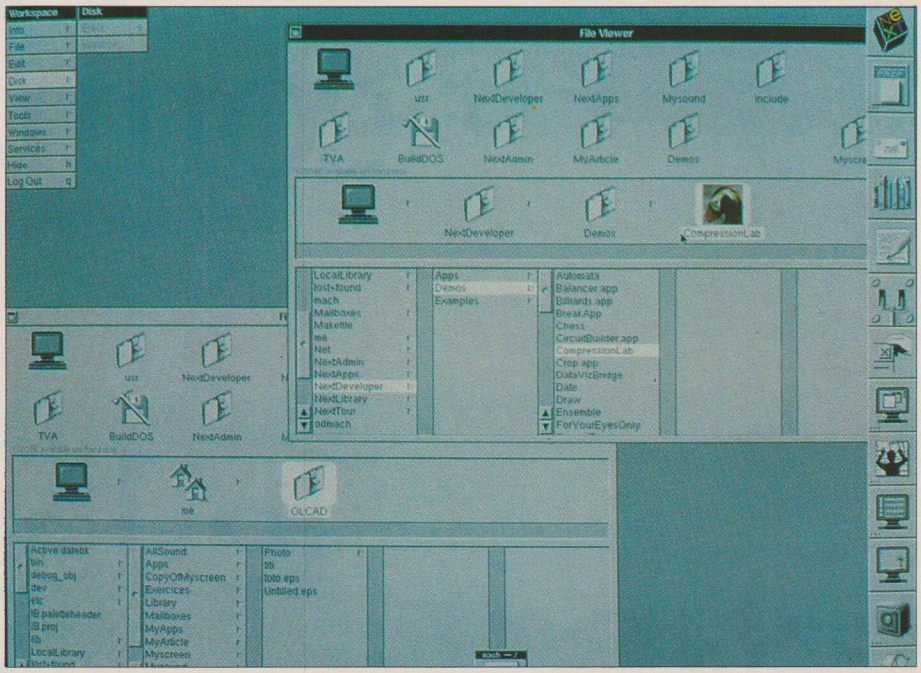
Une comparaison entre ces deux interfaces met en évidence que, contrairement à l'interface Mac, NeXTStep ne possède pas de barre de menus. Elle est remplacée, au sein du browser, par un menu principal qui ressemble, à s'y méprendre, à un menu classique.

Les fameux menus Mac font place, eux, à des sous-menus hiérarchiques. Ces derniers —équivalents des répertoires sous DOS— sont matérialisés par la présence d'une flèche en relief. Ils ont également l'avantage de comporter, à chaque niveau, le nom des applications correspondantes. Ce n'est pas le cas des menus sous Multifinder, dont on se demande parfois à quelle application ils appartiennent. Autre remarque : sur le Macintosh, les menus, déroulants, imposent à l'utilisateur de garder le doigt appuyé sur le bouton de la souris. Il n'a alors accès qu'au dernier menu déroulé. Sur



Développeurs, connaissez vous le langage qui vous permet de franchir toutes les frontières ?

Service lecteurs nº 17 (dernière page)



2 Browsers ouverts simultanément pour plus de confort d'utilisation.

NeXT, ces menus se déroulent horizontalement de façon automatique et restent statiques.

Ainsi, tous les fichiers, de tous les répertoires ouverts à partir de la racine, sont simultanément à la disposition de l'utilisateur. Ils ne se referment que si ce dernier en fait la demande, pour un confort d'utilisation accru. Il dispose de quantité de solutions ergonomiques qui améliorent la lisibilité de l'interface et sa facilité d'utilisation. Les menus, par exemple, s'éclairent lorsqu'ils sont sélectionnés.

Egalement, il ne peut exister plusieurs sous-menus d'un même niveau hiérarchique, ouverts simultanément sous le browser. Chaque sous-menu ouvert se referme automatiquement, dès lors que l'utilisateur choisit une autre commande ou reclique sur le menu parent. Un avantage non négligeable qui permet de laisser l'environnement de l'utilisateur propre et exempt de toute fenêtre superflue. Cependant, celui-ci devra se servir de deux browsers ouverts conjointement pour effectuer des opérations (de copie, par exemple), entre différents sous-sous-répertoires d'un même menu. Lorsqu'un sous-menu se trouve isolé, il acquiert automatiquement une case de fermeture.

A l'inverse, lorsque l'on déplace son menu parent, il le suit comme s'il était attaché avec des élastiques, produisant ainsi une animation surprenante. Tout comme les menus, les icônes et les boutons s'éclairent quand on les sélectionne.

Des icônes animés

Certaines icônes peuvent également s'animer. Ainsi, quand l'on copie une icône dans un dossier, l'icône dossier s'ouvre et semble l'avaler... Autre caractéristique, NeXTStep est une interface utilisateur entièrement reconfigurable. Mouse preferences, Keyboard, Display, general, Date & Time, startup, Password, Menu, Screens, Expert... sont autant de menus de préférences qui permettent à l'utilisateur de personnaliser et de modeler réellement son propre environnement.

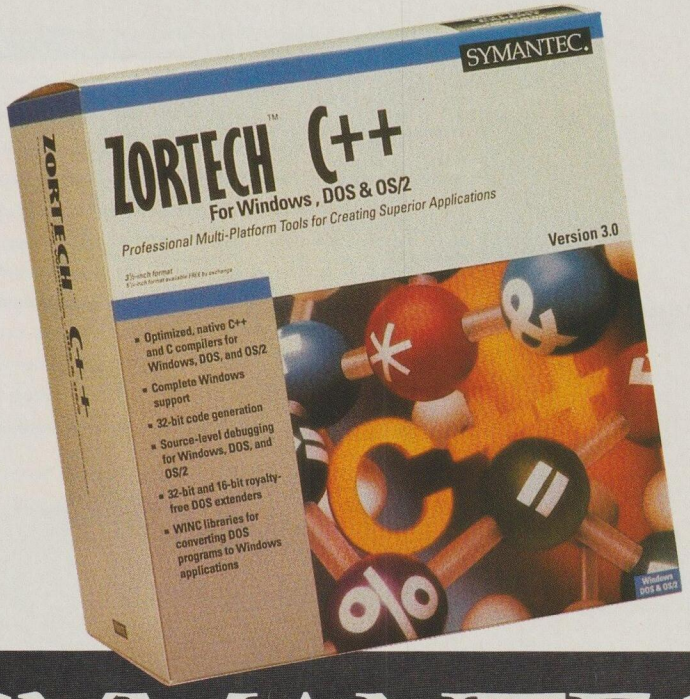
A ces nombreux menus vient s'ajouter la notion d'Inspecteur propre à NeXTStep. Les inspecteurs ressemblent aux fonctions propriétés disponibles sous MS-Windows, en version améliorée. Cette commande autorise l'affichage des caractéristiques de l'icône, du fichier ou du dossier sélectionnés, et de les modifier. Flexibilité, animation, jeux de lumière et soin artistique ont permis à NeXTStep de devenir une grande interface utilisateur ergonomique et agréable à manipuler.

Autant d'atouts que l'on retrouve au sein de l'ensemble des fonctions disponibles sous NeXTStep. Unix et Ethernet, par exemple, en profitent par son intermédiaire. Bon nombre de leurs fonctions ont ainsi été dotées d'une interface. Le Finder, lui, permet à tout utilisateur d'effectuer la recherche de fichiers, ou de dossiers, à travers son environnement. Il s'agit, ni plus ni moins, d'une implantation graphique de la commande Find disponible sous Unix.

Il en va de même pour la plupart des fonctions Unix, comme la commande Kill, dont l'équivalent graphique sous NeXTStep est l'opérateur Processes. NFS, également incorporé dans NeXTStep, apporte à tout utilisateur une facilité de déplacement dans les réseaux les plus compliqués. L'utilisateur a donc accès, à travers NeXTStep, à des fonctions complexes et souvent difficiles à mettre en œuvre sur d'autres plates-formes. De Librarian (véritable logiciel archiviste) à Display PostScript, de nombreuses applications s'intègrent entièrement dans NeXTStep. Display PostScript permet à l'utilisateur de travailler en Wysiwyg : il autorise l'affichage, à l'écran, d'images PostScript couleurs et de fichiers, tels qu'ils sortiront de l'imprimante. NeXTStep offre également aux utilisateurs l'occasion de communiquer.

Zortech

Le seul langage C++ parlé par tous les systèmes d'exploitation.



SYMANTEC.

Zortech C++ de Symantec, le compilateur sans frontières.
Zortech C++ est le seul compilateur C++ multi-plateforme, compatible avec tous les systèmes d'exploitation standards : DOS, Windows, OS/2, Unix et Mac. Si Zortech C++ est totalement compatible, il sait également utiliser à fond la puissance de chaque environnement. Chaque plateforme bénéficie ainsi de l'apport d'outils spécifiques.

Zortech C++, encore + de rapidité.
Avec sa capacité de compilation 32 Bits, Zortech C++ est le compilateur le plus rapide du marché. Sa puissance permet un portage rapide et facile des applications et vous assure une meilleure productivité.

Toujours + de facilité.
Zortech C++ de Symantec a été conçu pour tous les développeurs. Sa programmation Orientée Objets et sa capacité à exploiter pleinement les bibliothèques externes les plus courantes sur Mac et PC (compatible ANSI C, ANSI C++) offrent une souplesse et une facilité de programmation étonnantes.

Symantec, les logiciels de développement de référence.
En éditant THINK C, THINK Pascal (les deux logiciels de développement incontournables sur Macintosh) et Zortech C++, Symantec détient un savoir-faire unique qui en fait le partenaire de tous les développeurs,
Symantec s'affirme ainsi comme un grand éditeur, un grand éditeur de solutions.

Je désire recevoir une documentation complète sur :
☐ Zortech C++
☐ THINK C
☐ THINK Pascal

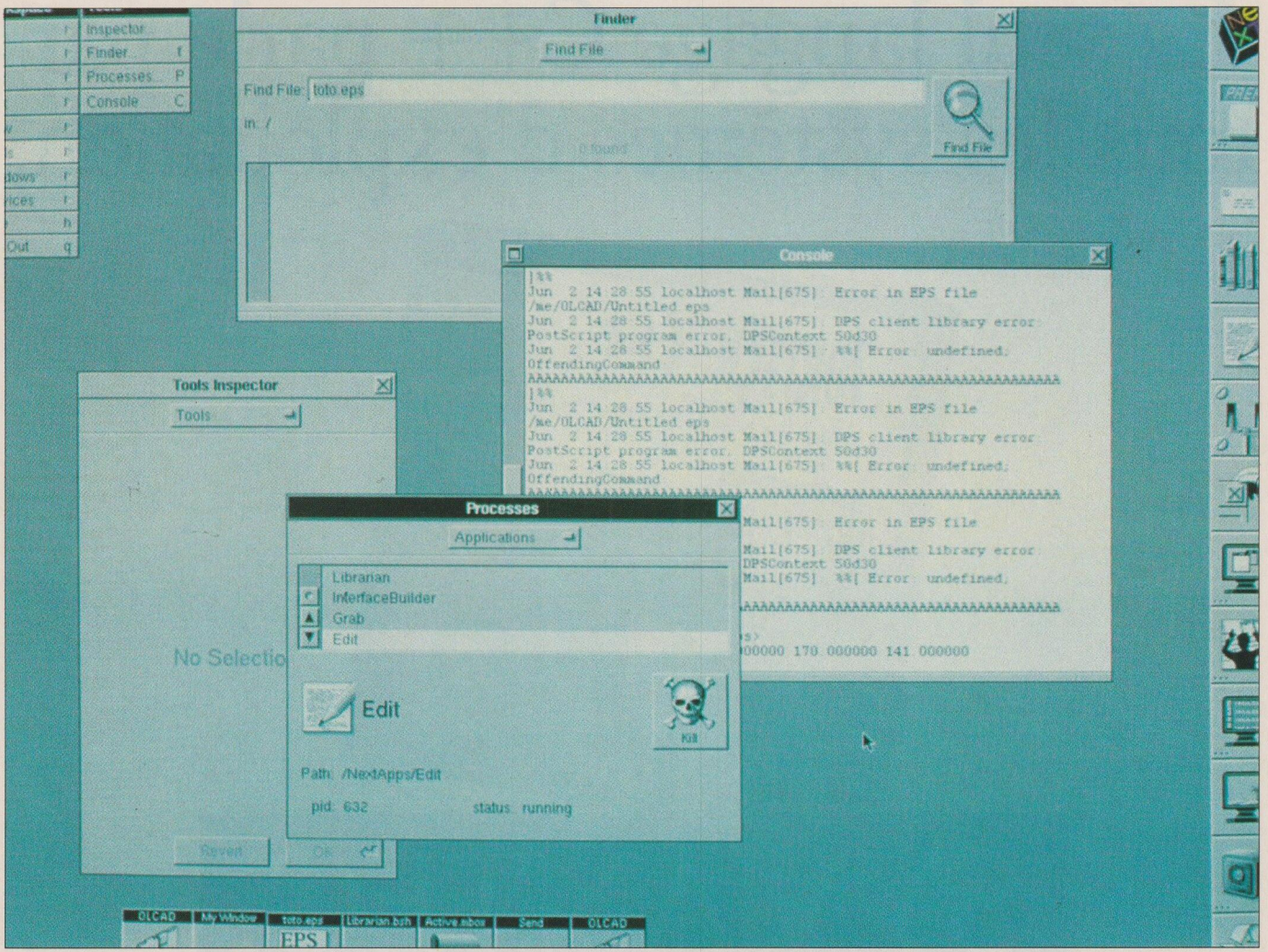
Nom _____ Prénom _____
Fonction _____ Société _____
Adresse _____
Code Postal _____ Ville _____
Tél. _____

A remplir soigneusement et à retourner à l'adresse suivante : Symantec France. Service clients. 31-35, rue Gambetta. 92150 Suresnes. Tél. : (1) 47 72 16 50. Minitel 3616 Symantec.

ÉDITEUR DE SOLUTIONS

SINGAPOUR

Service lecteurs n° 18 (dernière page)



Des commandes Unix enfin dotées d'une interface très conviviale.

Avec NeXTmail et NeXTfax, ils disposent d'un véritable système de messagerie électronique multimédia. Ils peuvent y intégrer du PostScript, du texte enrichi, des images couleurs, de la voix, de la musique ou encore n'importe quel document créé à partir d'une application existante sur le système NeXT. NeXTmail constitue en soi un véritable Workspace où l'utilisateur a la possibilité de copier des icônes — quel qu'en soit le contenu — et de lancer des applications! Mais NeXTStep est bien plus qu'une simple interface utilisateur graphique. C'est également un environnement de développement d'une qualité rarement atteinte. Là aussi, il s'avère utile de définir la nature d'un bon environnement de développement. Il doit permettre aux développeurs de créer très facilement des applications puissantes, en un minimum de temps. A cette fin, il doit :

• être extrêmement flexible et convivial. Cela implique qu'il dispose d'un grand nombre de fonctionnalités et d'une interface utilisateur graphique de grande qualité.

• offrir un maximum de générateurs de code automatiques. Comme celui de l'utilisateur, il lui faut pouvoir être réutilisable à tout moment. Le code généré doit donc être portable et répondre à un certain nombre de standards de l'industrie, afin que les mêmes algorithmes puissent s'exécuter sur différentes plates-formes.

Le développeur doit pouvoir disposer d'outils standards de développement, c'est-à-dire générant du code standard. Le code correspondant à une application se divise généralement en deux parties distinctes : les algorithmes, à l'arrière-plan, qui traitent les actions spécifiques et l'interface utilisateur au premier plan.

En résumé, un bon environnement de développement est donc nécessairement bâti autour d'un outil excellent, dans le but de créer des interfaces utilisateur et un ensemble complet d'outils standards de l'industrie, pour la génération ou le portage des algorithmes sous-jacents propres à l'application. Parce qu'il a été entière-

ment conçu et écrit en langage orienté objet et parce qu'il dispose d'un nombre d'applications considérable, NeXTStep, avec Interface Builder —un éditeur d'objets graphiques—, réunit tous ces éléments !

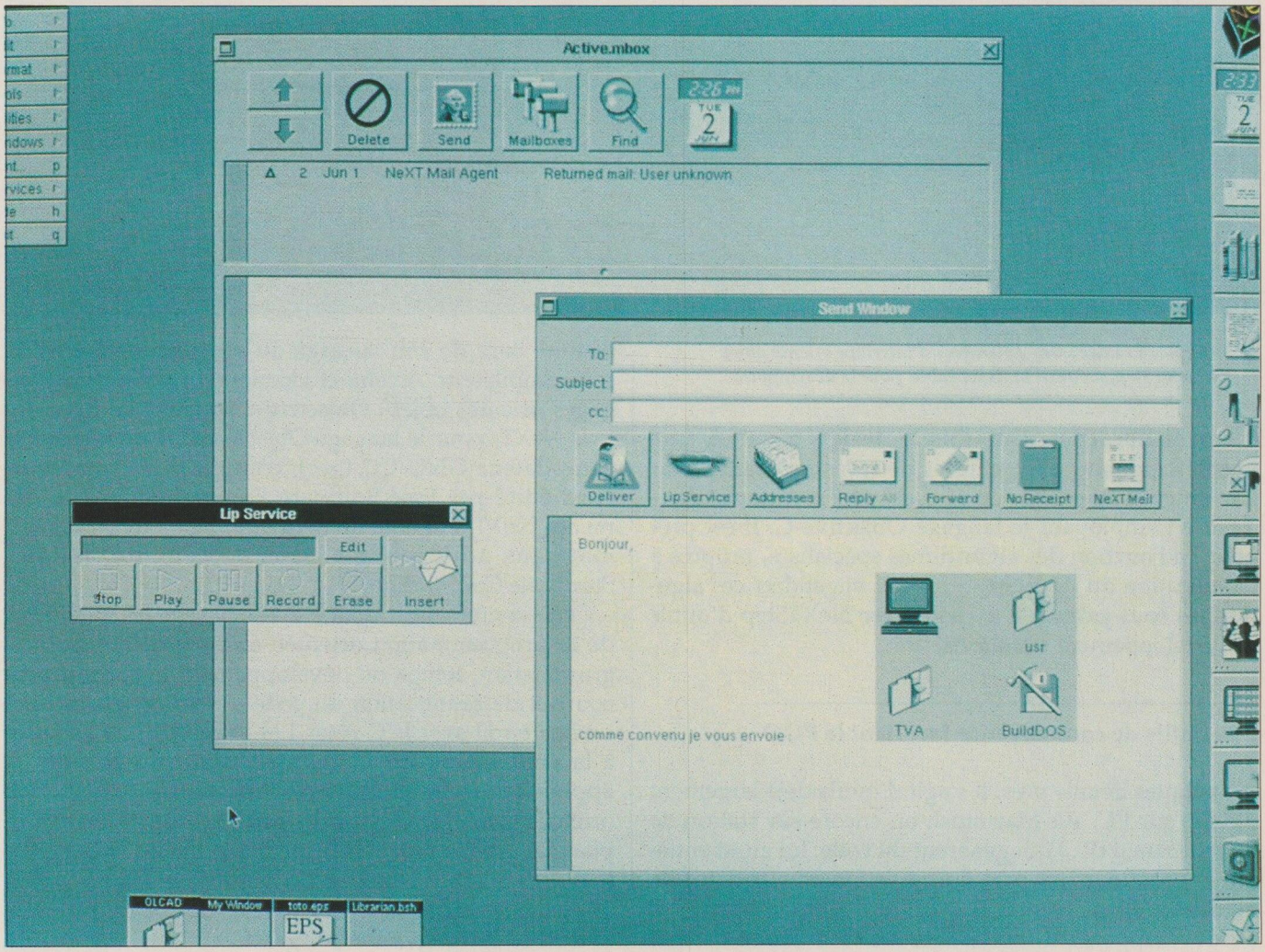
Il y a quelques années encore, les développeurs passaient 50% de leur temps à l'élaboration d'une interface. Sous DOS, le développeur ne disposait d'aucun outil efficace. Sous Macintosh, il était obligé d'écrire de nombreuses lignes de code pour effectuer le moindre test auprès d'un utilisateur potentiel. Avec DevGuide sur Sun ou HyperCard sur Macintosh, il devait souvent réécrire le code généré par l'outil de prototypage pour la création de l'interface utilisateur. Aujourd'hui, les éditeurs d'interfaces ne cessent de se développer sur une grande quantité de plates-formes.

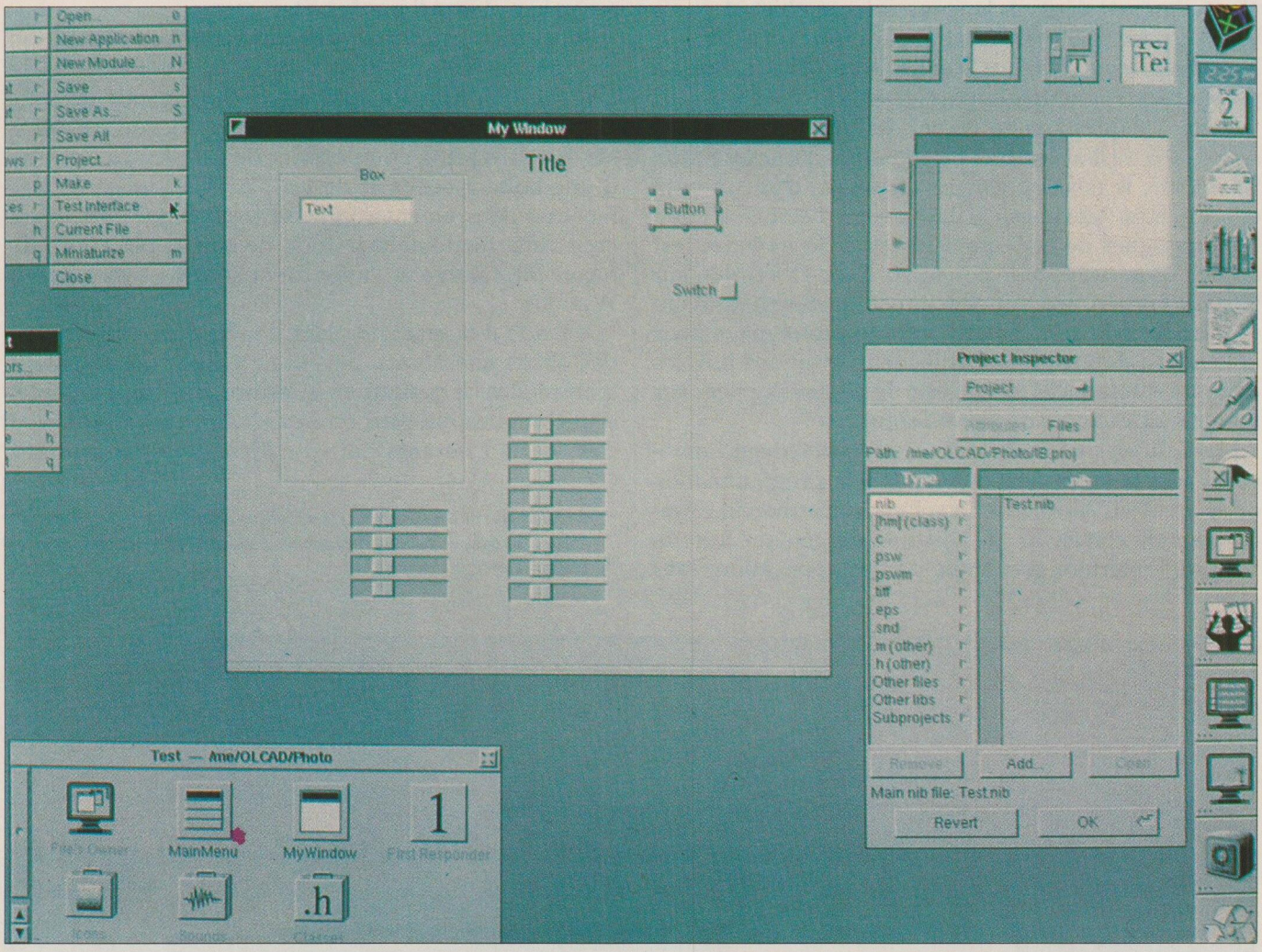
Mais aucun ne bénéficie d'un environnement homogène, adapté au développement et à la programmation orientée objet. Ainsi, Interface Builder permet au développeur de concevoir, de tester et de générer le code final de l'interface graphique de son application, sans écrire la moindre ligne de code.

Pour cela, l'environnement NeXT met à sa disposition un ensemble d'outils orientés objet, appelé «Application Kit», un équivalent 100% orienté objet de la ToolBox du Macintosh. Il lui suffit alors de faire glisser les différents composants de l'interface à partir d'une boîte à outils et de les placer à l'intérieur des fenêtres, dans un environnement graphique complet. Il crée ainsi facilement le look de son application, de façon interactive, à l'aide de la souris et en version Wysiwig.

A l'aide d'Interface Builder, l'utilisateur peut définir des objets inexistants, devant accomplir des fonctions bien réelles. Il peut alors effectuer le design complet d'une application sans être programmeur, en utilisant des objets existants dans l'Application Kit ou dans

NeXTMail : une messagerie électronique révolutionnaire ! Textes, icônes, images et même programmes peuvent être échangés entre utilisateurs.





Interface Builder : «le» générateur d'interface orienté objet de NeXTStep; un outil indispensable pour le développeur.

d'autres librairies. Une fois le design terminé, une simple commande lui permet de transformer sa création en code standard. Celui-ci est généré automatiquement en employant le langage Objective-C. Il est prêt pour l'adjonction des algorithmes spécialisés, propres à l'application du développeur. Pour engendrer ces algorithmes sous-jacents, il dispose avec NeXTStep d'outils de développement standards.

Des outils de connaissance facilitant la POO

A quelques détails près, il s'agit d'outils déjà largement utilisés sur PC, sur Macintosh ou encore sur station de travail (Sun, HP...) qui génèrent du code, lui aussi entièrement standard et prêt à être transféré sur d'autres machines. Ainsi, comme la plupart des environnements de développement modernes, NeXT a choisi C-Ansi comme base de son langage de programmation et de son compilateur. A celui-ci viennent s'ajouter deux langages orientés objet : Objective-C et C++. Le compilateur NeXT, pour le langage Objective-C, s'appuie sur le compilateur GNU CC (gcc), compilateur C standard Ansi édité par Free Software Foundation. A ces éléments, NeXT a rajouté un petit nombre de termes grammaticaux additionnels issus de ceux définis par Stepstone Corporation.

Cette combinaison donne à NeXT tous les avantages de la programmation orientée objet —facilité de programmation, temps de développement beaucoup plus court et de réutilisation du code— tout en maintenant un lien étroit avec le C-Ansi. Les développeurs habitués à la programmation C n'auront donc que très peu à apprendre le jour où ils passeront à une programmation orientée objet. NeXT fournit également un second compilateur orienté objet totalement standard : le compilateur C++ Ansi. A noter qu'Objective-C et C++ peuvent se mélanger et toujours être appelés de manière croisée. D'autres langages de programmation, tels que le

fortran 77, ISO pascal, Common Lisp, cobol-85 et SQL, sont également disponibles sur station NeXT. NeXTStep comporte, en plus, un grand nombre d'outils supplémentaires pour les développeurs.

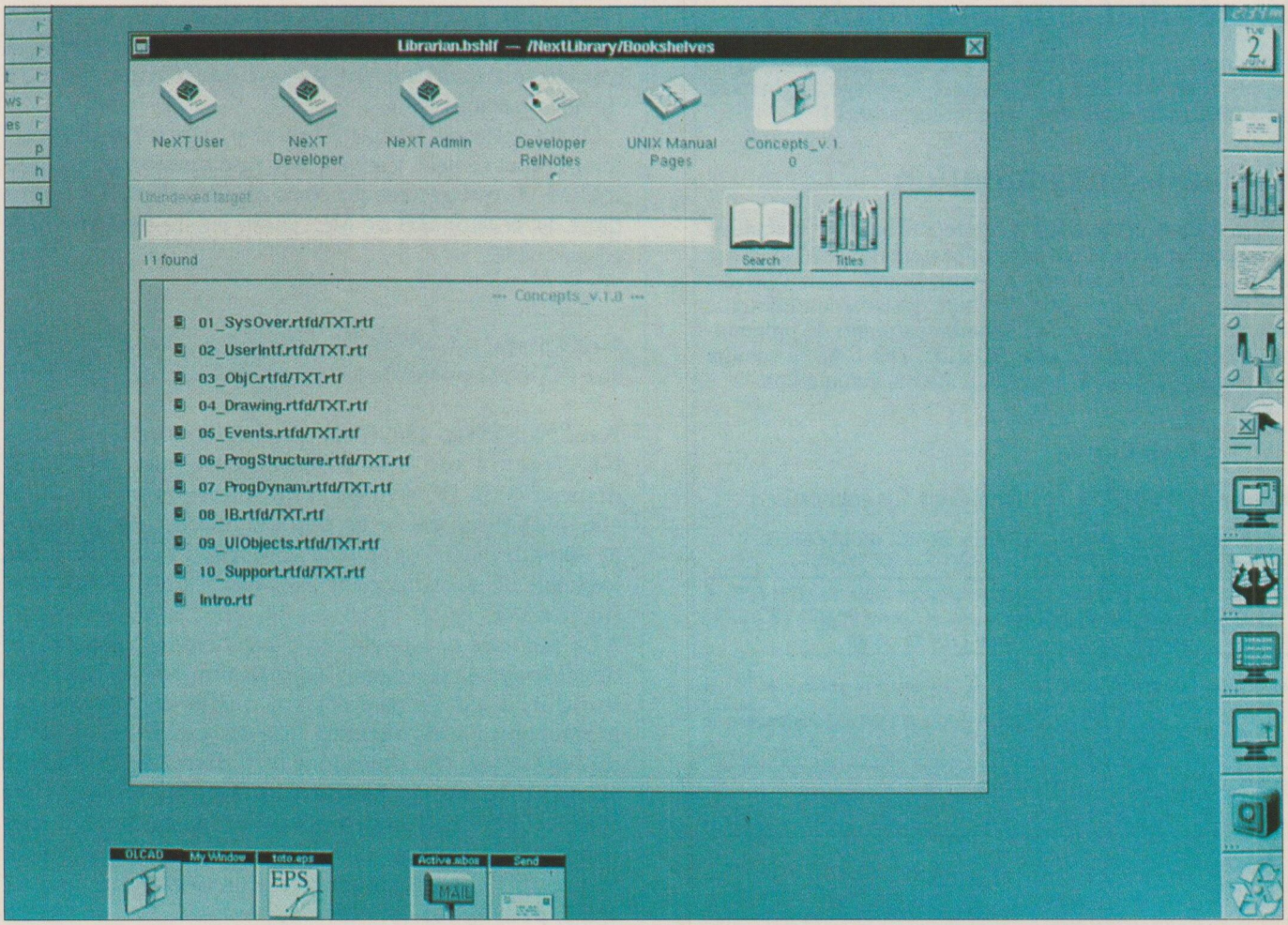
Chacun dispose d'une documentation en ligne omniprésente. DBKit, par exemple, est une toolkit qui autorise la gestion de transactions de bases de données et la création de son interface —à l'aide l'Interface Builder— tout en profitant de la programmation orientée objet. 3D Graphics Kit, basé sur le standard de Pixar RenderMan, offre, lui, un environnement orienté objet permettant d'obtenir un modèle unifié de rendu 3D.

Le développeur peut alors créer des applications 3D, ajouter des graphiques 3D à des applications existantes, les dessiner à l'écran et les imprimer. Intégrée à Objective-C, la fonction «Objets distribués» rend possible la génération d'applications de travail de groupe, en proposant l'envoi de messages entre objets sur différentes stations et au sein de diverses applications. Phone Kit, enfin, permet de développer facilement et rapidement des applications utilisant les possibilités de la téléphonie. Pour cela, il suffit d'employer les classes d'objets offertes par Phone Kit pour interfacer avec le téléphone et le RNIS. NeXTStep fonctionne sur tout ordinateur NeXT muni d'un minimum de 8 Mo de RAM et d'un disque dur de 100 Mo.

Chaque ordinateur NeXT est livré avec un CD-ROM contenant le logiciel NeXTStep d'une version donnée. Cette dernière, tout comme la configuration logicielle préinstallée à l'usine, dépend de la taille du disque dur. Sur disque de 100 et 200 Mo, il s'agit de NeXTStep version 3 Essentielle (prix US : 995 $ HT) qui comprend notamment les applications essentielles, les parties runtime des kits orientés objet, le logiciel système et les applications d'administration système. Les disques de 400, 600 Mo et de 1 Go contiennent, en plus, un jeu complet d'outils de développement : applications avancées, kits orientés objet au complet...

A souligner que le passage de la version Essentielle à la version complète de NeXTStep 3.0., coûtera environ 1 700F HT à tout détenteur d'une station NeXT équipée

Autre exemple de la convivialité et de l'efficacité de NeXTStep : Recherche dans la documentation, service «Librarian».



dBGEN 6.11

Générateur d'applications

DBASE IV, FoxPro, NANTUCKET

Générez vos applications dBASE sans programmer grâce à l'interface de paramétrage **QuickGEN**

- **Multi-fichiers** (en relation)
- **Multi-postes** (gestion des conflits)
- **Multi-écrans** (fixes ou déroulants)
- **Pop-menus** (1+2 niveaux d'appel)
- **Fenêtrage/sélection** (illimité)

➠ Simplicité, rapidité et souplesse

Les fichiers et index étant décrits dans un dictionnaire, il vous suffit pour définir un programme d'indiquer les fichiers en relation et la présentation formulaire ou déroulante pour chacun d'eux. Un éditeur WYSIWYG vous permet de tracer les écrans et positionner les champs en fenêtrage sur fichiers. Appels de fonctions et programmation dBase sont possibles.

Il ne vous reste plus qu'à choisir votre langage et générer les .PRG. Le code généré est simple, modifiable et exécutable sans "run time".

➠ Un résultat sans compromis

Enfin la minute de vérité est arrivée ! Vous découvrez une magnifique application avec menus déroulants, fenêtrage, pagination, gestion transparente du réseau ... du professionnel !

➠ Un outil sans limitation

Vous ferez vos maquettes en un temps record bien sûr, mais aussi vous irez jusqu'au bout de vos développements grâce à l'interface d'accès direct au dictionnaire. Ici vous entrez dans l'univers dBGEN, plus aucune limitation d'écriture, vous disposez de toute la puissance de l'atelier logiciel, (nombreux points de programmation, utilisation des IF, DO, WHILE, DO CASE, nombre illimité d'écrans, etc...). Avec documentation automatique

➠ Et la souris ...

En standard dans dBGEN, en option dans vos applications .

Marques déposées : NANTUCKET Corp, dBFAST (BUMBLEBEE), dBASE IV (ASHTON TATE), FoxPRO (FoxSOFTWARE).

Présentation les deuxièmes vendredi de chaque mois de 10 H à 16 H
HOTEL FRANTOUR 163 bis, avenue Clichy PARIS 17
Réservation : ALTI-SOFT 78 72 05 69

Je désire recevoir : une documentation ☐ une version de prise en main ☐
Ci joint un chèque de 150 F pour une prise en main

Sté _____
Adresse _____
_____
Nom _____

ALTI-SOFT 32, av. Félix FAURE 69007 LYON
Tél. :78 72 05 69

Service lecteurs n° 19 (dernière page)

d'un disque dur d'une capacité d'au moins 400 Mo.

NeXTStep a été conçu pour que les applications ne dépendent pas du système d'exploitation, ni de la plate-forme Hardware, et travaillent à un très haut niveau d'abstraction.

NeXTStep vers l'universalité

NeXTStep ne devrait donc pas tarder à apparaître sur plusieurs plates-formes. Chez NeXT, l'objectif principal est d'ailleurs de rendre NeXTStep le plus universel et le plus accessible possible, quelle que soit la machine utilisée. Une version de NeXTStep pour IBM RS 6000 existe déjà. IBM avait acheté les sources de NeXTStep 1.0, il y a environ trois ans, et celles de la version 2.0, il y a neuf mois. Big Blue ne devrait pas tarder à se doter de la version 3.0. Cependant, en ce qui concerne IBM, on voit mal comment il pourrait continuer de promouvoir un produit qui risquerait de faire de l'ombre à AIX.

Impression qui ne peut être que renforcée par l'accord relativement récent entre Visual Edge et IBM pour porter UIM/X sur IBM. Cet accord a donné naissance à AIC, un éditeur d'interfaces qui s'intègre parfaitement dans l'environnement de développement IBM.

Or on sait que Visual Edge s'apprête à commercialiser Quest Object Views C++, une version améliorée d'UIM/X, qui génère du code orienté objet. En attendant, la version 3.0 de NeXTStep existe déjà sur processeur 486.

NeXTStep sur PC, c'est possible !

Avec NeXTStep 486, il est désormais possible d'offrir NeXTStep à son micro-ordinateur, pour peu qu'il dispose d'un PC équipé d'un microprocesseur Intel 486. NeXTStep existe en deux versions : «utilisateur» et «étendue». La première comprend l'ensemble des logiciels nécessaires pour exécuter toutes les applications NeXT.

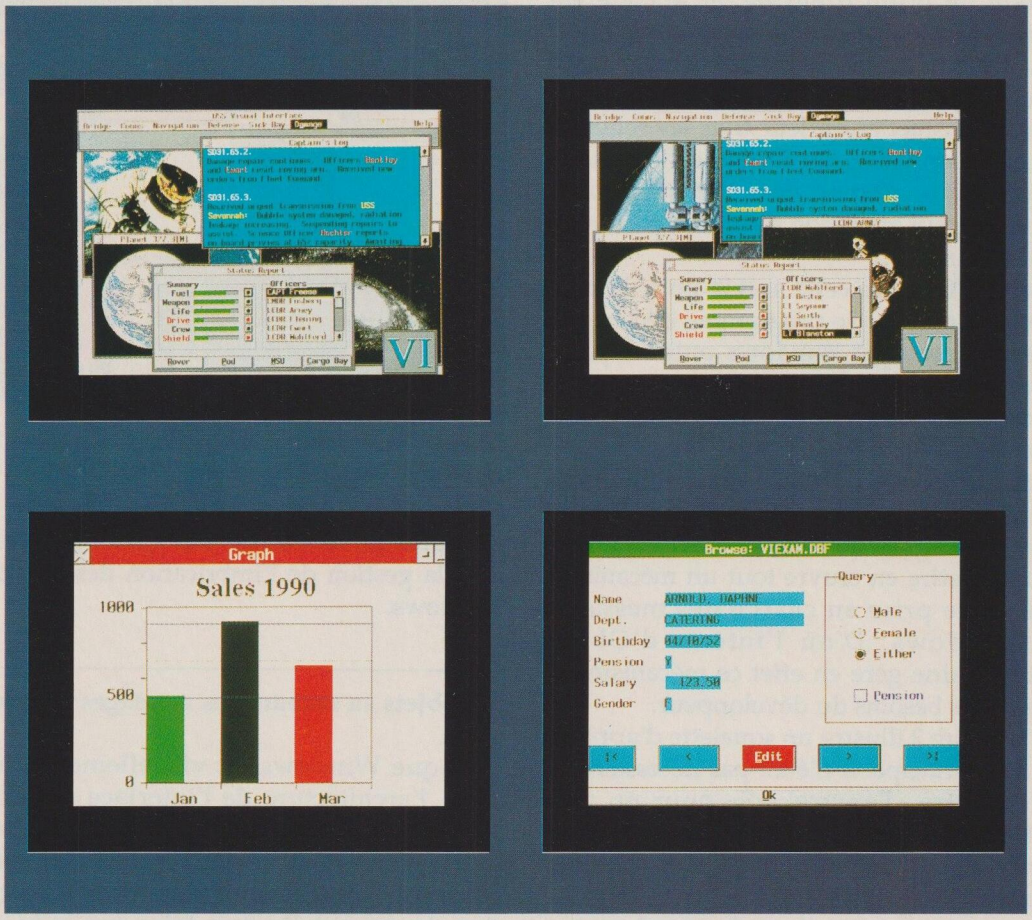
La seconde comporte, en plus, l'environnement de développement —dont l'Application Kit—, Interface Builder, ainsi qu'une série d'outils de développement, compilateurs et débogueurs. Les deux versions de NeXTStep 486 devraient être disponibles dans le courant du mois chez la plupart des revendeurs NeXT (Cub'X system...). Elles seront également distribuées sur support CD-ROM uniquement, aux prix respectifs de 995 $, pour la version utilisateur, et de 2 495 $, pour la version développeur. ❐

Visual Interface pour Nantucket 5

VI (Visual Interface) pour Nantucket 5 est une interface utilisateur graphique à la Windows qui vous offre pour la première fois les 3 fonctionnalités indispensables pour vos applications :

- Des fenêtres haute résolution avec les contrôles interactifs
- Un gestionnaire d'évènement intégré
- Des fenêtres, des objets graphiques, et des icônes qui s'intègrent parfaitement à Clipper 5

Les applicatifs peuvent être exécutés à partir du DOS 3 ou plus, ou lancés à partir de Windows.



Nantucket 5 et VI : les environnements de développement en mode graphique

Je désire recevoir la documentation concernant le produit Visual Interface pour Clipper

M, Mme, Mlle .....Prénom .....

Société .....Fonction .....Adresse.....

.....

Code Postal.....Ville.....Téléphone.....

PC Technologie - 4/4 bis, rue Bourgeoise - 92800 PUTEAUX
Tél : (1) 42 04 00 88 - Fax : (1) 42 04 69 80

PC TECHNOLOGIE

BITS PER SECOND

Service lecteurs n° 20 (dernière page)

LANGAGE

OBJECT WINDOWS
Une librairie objet complète et évolutive

JEAN-PAUL PRUNIAUX

Les développeurs d'applications sous DOS ont l'habitude de travailler de manière linéaire : le programme possède un point d'entrée, et le code est «déroulé» à partir de ce dernier. Si le programme effectue des accès au système ou manipule des fichiers, il fait appel à une ou à plusieurs fonctions des bibliothèques utilisées, celles-ci transmettant à leur tour les requêtes aux fonctions correspondantes du DOS ou du Bios. La tâche du développeur se résume alors à traiter le code de retour pour savoir si l'opération s'est bien passée ou s'il s'est produit une erreur.

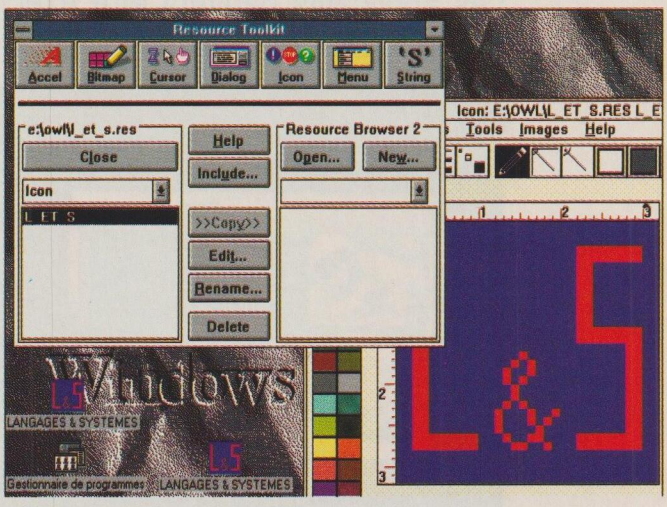
Cette démarche diffère de celle suivie pour la programmation sous Windows : le programme déclare ses ressources à l'environnement Windows puis lui passe la main. Le processus se trouve alors inversé : c'est le système qui, aléatoirement, rend la main au programme pour lui signaler ce qui vient de se passer.

Un programme développé dans un langage traditionnel (C, par exemple) qui affiche «Hello World» nécessite ainsi environ 150 lignes de code (250 avec les commentaires). Il faut mettre en œuvre tout un mécanisme de déclaration et de prise en charge des messages envoyés par Windows. D'où l'intérêt d'Object Windows. Cette librairie gère en effet ce mécanisme et peut évoluer selon les besoins du développeur.

Le listing de la figure 2 illustre un squelette d'application bâti grâce à ce principe. Il n'était pas nécessaire de créer un objet fenêtre «FenêtreL&S» pour un programme de cette taille, mais cela s'avère utile pour un squelette d'application, car il est possible d'en modifier les caractéristiques.

Nous ne traiterons pas, dans cet article, tous les paramètres passés aux diverses fonctions de cet exemple. Celui-ci illustre la simplification apportée par Object Windows. A noter que la procédure WinMain est l'équivalent pour Windows du traditionnel point d'entrée en C : main (...).

Les paramètres passés par Windows permettent d'identifier l'application, une éventuelle utilisation simultanée de l'application, une ligne de commande et un statut de fenêtrage. Les autres paramètres concernent la gestion de l'imbrication des objets dans Object Windows.



Le menu principal du White Water Resource Toolkit et, en arrière-plan, l'éditeur d'icônes.

Les objets au secours des messages Windows

Bien que Windows ait visuellement une apparence objet, l'architecture de l'interface de programmation avec cet environnement graphique n'a rien à voir avec la programmation orientée objet. Windows rend la main à l'application de manière aléatoire pour lui signaler ce qu'elle doit faire par le biais d'un code (déplacement de la souris, etc.). D'autres paramètres sont passés avec ce code afin de qualifier l'événement, et le tout constitue un message Windows.

L'application doit fournir à Windows une fonction permettant de recevoir tous ces messages. Cette routine redirige alors les messages vers la fenêtre active de l'application, et une analyse du message doit être faite au moyen d'un «switch» gigantesque qui gère le traite-

Le pack «Borland C++ and Application Frameworks» contient un compilateur C++, des outils de développement, de gestion de ressources pour Windows et une version C++ de Turbo Vision et d'Object Windows. Cette librairie d'objets met en place tous les mécanismes dont une application a besoin, faisant tomber à une dizaine d'instructions la taille du programme. Grâce aux mécanismes d'héritage de la programmation orientée objet, ce moteur de base peut également être transformé, enrichi et adapté en fonction des besoins du programmeur.

ment en fonction du message reçu. Tout ce mécanisme est pris en charge par Object Windows qui le traite de manière transparente.

L'objet application (ou son dérivé, «AppliL&S» dans notre exemple) récupère tous les messages Windows et procède au dispatching sur l'objet «fenêtre» concerné. Chaque objet de type fenêtre comprend des routines par défaut pour traiter ou non ces messages. Un objet fenêtre contient également un certain nombre de comportements types gérés par Object Windows. Enfin, grâce au mécanisme d'héritage, le programmeur peut déclarer une nouvelle méthode ou une fonction membre pour gérer et personnaliser n'importe quel message Windows ou tout comportement type.

Les méthodes virtuelles dynamiques

Le listing de la figure 3 illustre comment, grâce à la programmation orientée objet, il est possible de récupérer n'importe quel événement Windows ou Object Windows. Pour intercepter ces deux événements, des déclarations différentes ont été utilisées. «CanClose» est une méthode virtuelle, comme celle disponible sous Turbo C++ et Turbo Pascal. Avec Borland C++ et Turbo Pascal pour Windows, il existe une nouvelle possibilité, les «fonctions de réponse à des messages», qui peuvent être aussi être appelées méthodes virtuelles dynamiques.

En Turbo C++, le polymorphisme est géré grâce aux méthodes virtuelles. Un objet «Parent» possède une méthode virtuelle «DessineToi()»; un objet «Enfant» possède sa propre méthode virtuelle «DessineToi()», cette dernière faisant un travail différent. Cependant, les deux méthodes virtuelles «Parent::DessineToi()» et «Enfant::DessineToi()» possèdent le même nom et la même déclaration. Cela permet à un programme manipulant un objet de dire «Objet.DessineToi()», sans savoir s'il a affaire à un «Parent» ou à un «Enfant». L'ensemble est géré en interne par le compilateur grâce à une table de saut indirect, la table des méthodes virtuelles, dans laquelle il reporte l'adresse de la méthode «Parent::DessineToi()» ou «Enfant::DessineToi()».

Borland C++ accède aux méthodes virtuelles dynamiques par un autre mécanisme. Celles-ci ne sont plus identifiées par un nom et par une adresse connus à la compilation, mais par un nombre ou par un numéro. La table des méthodes virtuelles dynamiques contient alors le numéro et l'adresse de chaque méthode.

Object Windows utilise cette propriété pour assigner des méthodes virtuelles à chaque message Windows. La déclaration «virtual void WMLButtonDown (RTMessage Msg) = [WM_FIRST + WM_LBUTTONDOWN];» permet ainsi d'assigner à la méthode gérant un clic de souris le numéro calculé grâce aux constantes

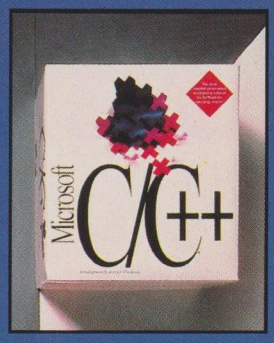
FIGURE 1

```
/*-----------------------------------
   Définition de l'icône de la
   fenêtre L&S
 -----------------------------------*/

class TFenetreLS : public TWindow
    {
    public:
    TFenetreLS (PTWindowsObject
 AParent, LPSTR ATitle)
       : TWindow (AParent, ATitle)
       {};
    virtual void GetWindowClass (WNDCLASS
_FAR & AWndClass);
    virtual BOOL CanClose();
    virtual void WMLButtonDown (RTMessage
Msg) =
       [WM_FIRST + WM_LBUTTONDOWN];
    };
void TFenetreLS::GetWindowClass (WNDCLASS
_FAR & AWndClass)
    {
    TWindow::GetWindowClass (AWndClass);
    AWndClass.hIcon = LoadIcon
(GetApplication()->hInstance, "L&S");
    };
```

40.18.10.70

!!! S.O.S. DEVELOPPEURS !!!

PROMO DU MOIS

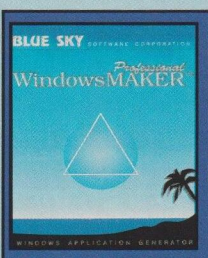


Microsoft C++
2099 Frs HT
(soit 2490 Frs TTC)

COMPILATEURS

LANGAGE C ++	Frs HT	Frs TTC
MS C++ 7.0	2099	2490
Borland C++ 3.0 Fr	2317	2748
Borland C++ / Application Frame 3.0 Fr	3477	4124
Turbo C++ 2nde Edition Fr	647	767
Turbo C++ Windows Fr	930	1100
Turbo C++ & Turbo Vision Fr	1037	1230
Zorthec C++ Compiler	2487	2950
Zorthec C++ Development	3766	4466
BASIC		
MS Basic PDS 7.1 US	2594	3076
Quick Basic 4.5 Fr	644	763
LANGAGE C		
Quick C 2.5 Fr	644	763
Quick C / Windows Fr	1154	1369
MS C 6.0a	2198	2607
PASCAL		
Turbo Pascal 6.0 Fr	927	1099
Turbo Pascal 6.0 Professional Fr	1857	2200
Turbo Pascal / Windows Fr	1472	1746
MS Pascal PDS 4.0	2594	3076
Quick Pascal 1.0 Fr	644	763

Support technique par les filiales françaises de chaque concepteur.



Windows Maker Professional

Développez vos applications Windows en cliquant sur votre souris, Windows Maker Pro génère directement tout le code C (documenté). Economisez des jours de développement et gagnez en maintenabilité. Testez le Look & Feel et effectuez les changements sans recompiler. Déjà Best Seller USA des développeurs C, la nouvelle extension MFC++ (module séparé) fait de Windows Maker Pro la référence des développeurs Microsoft C++ 7.0.

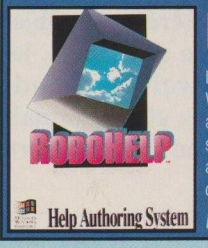
4900 Frs HT (soit 5811 Frs TTC)

!!! PROMOTION !!!
Windows Maker Pro & MFC++ & MS C++ 7.0
7390 Frs HT (soit 8765 Frs TTC)

Magic Fields

Toute la validation des champs de saisie est contrôlée (date, numérique...) par Magic Fields qui génère directement le code C. De plus, Magic Fields vous donne la possibilité d'offrir un look "NeXT" avec des reliefs à vos applications.

Nouveau !!!



RoboHELP

La génération des fichiers d'aide Windows en Hypertext n'a jamais été aussi simple et rapide. RoboHELP s'appuie sur Windows Word pour vous aider à réaliser vos fichiers Hypertext en cliquant sur la souris.

Nouveau !!!

BIBLIOTHEQUES / OUTILS SPECIAL WINDOWS

OBJET TABLEAU	Frs HT	Frs TTC
Soft Fields	2108	2500
Object Table C (avec source)	2943	3490
OBJET GRAPHIQUE		
HALO Image File Format Lib	2192	2600
Soft Chart	2951	3500
TIFF, GIF, PCX, Image : SDK	*nous consulter*	
OPTIMISATION		
MemShield 1.0	3356	3980
OptiMem (C & C++)	3024	3587
WindExe	2782	3300
ACCES AU SGBD		
Q+E DataBase Library Windows	2656	3150
QEVB (SPECIAL VISUAL BASIC)	1400	1660
Code Base 4.5	2867	3400
DataLib	3583	4250
COMMUNICATION		
Distinct TCP/IP 2.0a	*nous consulter*	
Chameleon TCP/IP 3.0	*nous consulter*	
TEST		
Microsoft Test	1850	2194
SQA Robot 1.1	*nous consulter*	
PORTAGE WINDOWS / DOS / OS/2		
MEWEL	2980	3535
WinPort	3200	3795
GENERATEUR D'INTERFACE		
Windows Maker Professional 3.10	4900	5811
3 in 1 C	1813	2150

VALIDATION DE SAISIES		
Magic Fields 1.0	**2800**	**3321**
Dialog Coder 3.0	4123	4890
EDITEUR / DEBUGGER		
ICE/W	*nous consulter*	
Codewrite	2437	2890
Periscope 32 Debugger Windows	2487	2950
DIVERS		
RoboHELP 1.0	**3800**	**4507**
ReportRight	*nous consulter*	
Drover's Toolbox	2487	2950
Kansman Pack	4115	4880
(avec source : table / statut / ribbon / toolbox)		
GENERATEUR D'APPLICATION		
Object Vision 2.0	3187	3780
Infiné/2 Windows	13406	15900
Spinnaker (Windows / Mac)	*nous consulter*	
Realizer	*nous consulter*	
SPECIAL VISUAL BASIC		
Visual Basic	1154	1369
Professional ToolKit for Visual Basic	1533	1818
Visual Basic Lib. & SQL SDK	2312	2742
QEVB	1400	1660
Visual Architect	1927	2286
VB Assist	1710	2028
DDE Tools	*nous consulter*	

BIBLIOTHEQUES / OUTILS SPECIAL C++

GENERATEUR D'INTERFACE	Frs HT	Frs TTC
Windows Maker Professional & MFC++	**5699**	**6759**
Extension MFC++ pour WMP	3377	4005
3 in 1 C++	1644	1950
Win ++	1644	1950
DIVERS		
Dynamic Object C++	2066	2450
RTLink Plus	3693	4380
Tools.h ++ (avec source)	2266	2687
Brief C++	*nous consulter*	
CodeBase ++	1855	2200
Greenleaf Comm ++	1813	2150
Object Table C++ (avec source)	2943	3490
POET (Base de Données Objet sous Windows)	*nous consulter*	

BIBLIOTHEQUES / OUTILS C & PASCAL

POUR LANGAGE C	Frs HT	Frs TTC
C Tools +	1096	1300
GreenLeaf Superfunctions 1.1	2066	2450
Polylibrarian II	1981	2350
HALO Professional	3035	3600
Plink 86 +	2766	3280
Blinker	1813	2150

POUR PASCAL		
Turbo Vision Dev ToolKit	1433	1700
Object Professional 1.1	1518	1800
Turbo Professional 5.1	1180	1400
Turbo Magic 2.02	1813	2150
Turbo Analyst 6.0	1180	1400

SGBD

SGBD	Frs HT	Frs TTC
SQL Server V 1.11 10 postes	**11992**	**14223**
Oracle	*nous consulter*	
Paradox 3.5 Fr	4874	5780
Paradox Engine 2.0 US	3497	4147
Fox Pro Mono 2.0 Fr	5852	6940
Fox Base + 2.1 Fr	5299	6285

• 28, Bd de Strasbourg • 75010 PARIS • Tél. 40.18.10.70 • Fax 40.18.10.77 •

• Merci de bien vouloir préparer votre commande, format disquette et version logiciel, adresse de livraison et adresse de facturation, avant de nous appeler •
• Règlements par chèque à la commande à l'ordre de OSER ou par Contre-Remboursement (avec supplément) • Autres, nous consulter •
Frais de port France Métropolitaine : 80 F (95 F TTC) pour commande inférieure à 1200 F HT, 150 F (177,90 F TTC) pour commande de 1200 à 5000 F. Nous consulter pour commande supérieure

Service lecteurs n° 21 (dernière page)

EQUIP'CONTACT 40 22 09 23 R.C. PARIS B 338 013 154 (88B 02167)

TOUS NOS PRIX SONT EN FRANCS FRANÇAIS ET SONT SUSCEPTIBLES DE MODIFICATIONS SANS PRÉAVIS • DANS LA LIMITE DES STOCKS DISPONIBLES •

FIGURE 2 : EXEMPLE «MINIMUM»

```
/*-------------------------------------*/
/* Programme 'Minimum' créé avec
   Object Windows

   Jean-Paul Pruniaux le 2.11.91      */

/*-------------------------------------*/

#include <owl.h>

/*-------------------------------------*/
/* Un objet de type Window contient
   toutes les caractéristiques par
   défaut d'une fenêtre, nous
   construisons ici un objet
   'FenêtreLS' qui hérite de toutes
   ces caractéristiques, celles-ci
   pourront être modifiées au gré du
   programmeur                        */
/*-------------------------------------*/

class TFenetreLS : public TWindow
    {
    public:
    TFenetreLS (PTWindowsObject AParent,
    LPSTR ATitle) :
       TWindow (AParent, ATitle)
        {};
    };
/*-------------------------------------*/
/* Un objet de type Application
   contient toutes les caractéristiques
   par défaut nécessaires à l'exécution
   d'un programme Windows.
   Nous construisons ici un nouvel
   Objet 'AppliLS' qui hérite de ces
   caractéristiques, et nous modifions
   une fonction d'initialisation qui
   construira une fenêtre             */
/*-------------------------------------*/

class TAppliLS : public TApplication
    {
    public:
    TAppliLS (LPSTR AName, HANDLE
     hInstance, HANDLE hPrevInstance,
     LPSTR lpCmdLine, int nCmdShow) :
       TApplication (AName, hInstance,
       hPrevInstance, lpCmdLine,nCmdShow)
        {};
    virtual void InitMainWindow();
    };

void TAppliLS::InitMainWindow()
    {
    MainWindow = new TFenetreLS (NULL,
Name);
    }
/*-------------------------------------*/
/* Programme principal appelé par
   Windows, celui initialise l'objet
   'Application' et lui donne la main
/*-------------------------------------*/
int PASCAL WinMain (HANDLE hInstance,
HANDLE hPrevInstance, LPSTR lpCmdLine,
int nCmdShow)
    {
    TAppliLS AppliLS ("Programme
    L&S", hInstance, hPrevInstance,
    lpCmdLine, nCmdShow);

    AppliLS.Run ();
    return AppliLS.Status;
    }
```

WM_FIRST + WM_LBUTTONDOWN. La première de ces constantes identifie le premier des messages Windows et la deuxième le numéro relatif de l'événement «clic sur le bouton gauche». Ces constantes, ainsi que bien d'autres gérées par Object Windows, sont déclarées dans le fichier d'en-tête «OWL.H».

Outre une grande souplesse d'utilisation et de programmation, cette organisation apporte un avantage énorme : elle permet un apprentissage étape par étape de l'environnement Windows.

Les ressources des programmes Windows

Une programmation pour Windows ne peut se contenter d'aligner du code pour prendre en main les divers événements transmis par Windows. Il faut aussi pouvoir intervenir dans une fenêtre. Les programmes Windows sont organisés en deux parties distinctes : le code (les instructions) et les ressources (menus, boîtes de dialogue, icônes, dessins point par point...). Windows fournit des outils pour gérer ces ressources. Un exécutable pour Windows contient donc du code et des ressources. Les outils de développement pour Windows de Borland sont livrés avec des outils permettant de créer ou de modifier des ressources, tel que le White Water Resource Toolkit. Cet outil, bien que gourmand en puissance machine, est très souple et permet même de modifier les ressources déjà existantes dans un fichier exécutable. Les ressources sont de plusieurs types :

- Les icônes, une fenêtre ou tout autre élément identifiable par Windows. Une même icône peut posséder plusieurs définitions, selon la qualité de l'écran graphique et le nombre de couleurs affichées simultanément. Windows sélectionne la plus appropriée.
- Les dessins point à point (bitmap) dont on peut confier l'affichage à Windows.
- Les curseurs de la souris. La forme de curseur peut changer selon la tâche effectuée par l'utilisateur du programme.
- Les menus. Créés sous forme de ressources, ils peuvent être gérés automatiquement par Windows dès qu'une option est sélectionnée. Windows retourne un code identifiant la commande concernée. Outre le fait que le programmeur n'a pas à générer de module de

gestion de menu, cela facilite également les tâches de traduction. Un menu peut être ainsi corrigé sans avoir à recompiler le programme.

• Les boîtes de dialogue. Le principe est identique à celui des menus, les éléments gérés étant ici des boîtes de dialogue.

• Les chaînes de caractères.

• Les accélérateurs, lesquels permettent de naviguer plus rapidement dans un programme.

Une application Object Windows contient déjà un certain nombre d'icônes (point d'interrogation, signe d'alerte...) et de curseurs prédéfinis. La première des ressources mise en œuvre est généralement un menu. La documentation explique clairement comment procéder. La figure 1 illustre la déclaration de l'icône faite par l'application à Windows. Il suffit de surcharger la méthode «GetWindowClass()» qui définit un certain nombre d'attributs par défaut transmis à Windows. Le champ «hIcon» de la structure est modifié pour désigner l'icône «L&S» des ressources.

Le White Water Resource Toolkit

Plusieurs méthodes permettent de mettre en œuvre des ressources et de les attacher à un module exécutable. Une manière simple consiste à faire comme avec Turbo Pascal pour Windows. Plutôt que de définir toutes les ressources une à une, puis de les assembler grâce à un script fourni au compilateur de ressources, on peut directement réaliser un fichier de ressources (fichier d'extension .RES) avec le White Water Resource Toolkit.

Ces fichiers .RES sont l'équivalent des fichiers objet, .OBJ. Les premiers concernent les ressources Windows, les seconds le code compilé correspondant au programme. Une fois le fichier .RES réalisé ou modifié avec le White Water Resource Toolkit, il suffit de l'intégrer dans le projet. A titre d'exemple, notre projet contient les éléments suivants :

" LS.CPP
" IPC.RES
" OWL.LIB
" CTRLL.LIB
" \BC\OWL\LIB\OWL.DEF

Une bibliothèque d'objets

Non seulement Object Windows constitue une bibliothèque fournissant une organisation et une structure de base simples pour la programmation pour Windows, mais elle comporte aussi un certain nombre d'objets préfabriqués tels qu'une calculatrice, une fenêtre de texte, un éditeur de fichiers ASCII, des boîtes de dialogue standard (ouverture de fichiers, par exemple), une fenêtre avec des ascenseurs, ainsi que tous les petits gadgets dont on peut avoir besoin dans la réalisation d'une boîte de dialogue. On y trouve également des fenêtres respectant l'interface de document multiple (MDI) permettant à une fenêtre mère de contenir plusieurs fenêtres filles, comme peuvent le demander des applications importantes. Signalons également que toutes les fonctions d'interface (API) avec Windows sont accessibles avec les librairies de Borland C++.

En conclusion, le compilateur C++, ses librairies et Object Windows permettent de s'attaquer à n'importe quel problème de programmation sous Windows. Pour couronner le tout, Object Windows est livré avec une grande quantité d'exemples. ❐

FIGURE 3

```
/*-----------------------------------*/
/* Définitions de deux méthodes, l'une
   interceptant le message Windows
   'On a appuyé sur le bouton gauche de
   la souris', l'autre interceptant la
   méthode Object-Windows vérifiant si on
   a le droit de fermer une fenêtre.
/*-----------------------------------*/
class TFenetreLS : public TWindow
    {
    public:
    TFenetreInfoPC (PTWindowsObject
    AParent, LPSTR ATitle): TWindow
    (AParent, ATitle)
       {};
    virtual BOOL CanClose();
    virtual void WMLButtonDown (RTMessage
    Msg) =[WM_FIRST + WM_LBUTTONDOWN];
    };

/* Interception d'un message souris de
   Windows, Affichage d'une boîte de
   dialogue (commentaire)              */
/*-----------------------------------*/

void TFenetreLS::WMLButtonDown(RTMessage)
    {
    MessageBox (HWindow, "Bouton gauche",
"Test de message", MB_OK);
    }

/* Interception de la vérification de la
   destruction de la fenêtre 'Boîte de
   dialogue' fictive demandant si on peut
   le faire                            */
/*-----------------------------------*/
BOOL TFenetreLS::CanClose()
    {
    return MessageBox (HWindow, "Faut-il
sauvegarder ?","Le dessin est modifié",
MB_YESNO | MB_ICONQUESTION) == IDNO;
    }
```

Vous connaissez l'une de nos casquettes, découvrez les autres

EDITEUR DE LOGICIEL • FORMATION
INGENIERIE LOGICIELLE • RECHERCHE

Depuis sa création, NSL innove dans les domaines de l'interface utilisateur par le choix de technologies gagnantes: UNIX, le système de fenêtrage X et OSF/Motif. NSL a développé son savoir-faire autour de quatre axes: la conception de logiciels, la formation, l'ingénierie logicielle et la recherche.

Nos progiciels (XFM2, Wx2, WISh2, wi) sont leaders dans leur domaine avec entre autre le choix par Atari US Corp. d'en équiper ses stations UNIX. Numéro 1 en France dans la formation au système de fenêtrage X, à Motif et C++, NSL est aussi centre de formation PostScript agréé ADOBE. Notre expérience en ingénierie logicielle nous a valu d'être retenus pour des projets de très haute technicité, tel que le développement de l'interface graphique d'un logiciel de PAO.

L'ensemble de ces compétences permet aujourd'hui à NSL de proposer des produits et prestations de qualité. NSL est membre de l'OSF.

NSL
NON STANDARD LOGICS

57/59, RUE LHOMOND 75005 PARIS TEL: (1) 43 36 77 50 • FAX: (1) 43 36 59 78

UNIX, X ET OSF/Motif sont des marques déposées.

Service lecteurs n° 22 (dernière page)

WINDOWS

WINDOWS 3.1
Bienvenue à TrueType, à l'OLE et au nouveau DDE

CHARLES PETZOLD

Microsoft a fait de son mieux pour apaiser l'angoisse du «point zéro» en procédant à un considérable test bêta pour l'environnement graphique Windows 3.0 et pour son système d'exploitation MS-DOS 5.

Cette démarche semble avoir porté ses fruits : ces deux produits sont entrés sur le marché dans un bon état et ont séduit des millions d'utilisateurs. Mais la firme de Redmond ne s'est pas arrêtée là. Le passage en mode protégé — sans doute l'amélioration la plus marquante de Windows 3.0 — n'était qu'un premier pas. Grâce au mode protégé, la mémoire est disponible pour bien d'autres implémentations, et une mise à jour s'avérait inévitable.

Profiter de TrueType

Vous avez peut-être entendu dire que Windows 3.1 était, en simplifiant, «Windows 3.0 plus TrueType plus quelques corrections mineures». Si vous l'avez cru, détrompez-vous. Windows 3.1 représente la plus grande amélioration de l'API Windows depuis la création de cet environnement.

Windows 3.1 possède un tiers de fonctions de plus que Windows 3.0. Les extensions Multimedia et Windows for Pens portent le total à plus de 1000! Le passage de Windows 2.x à 3.0 était facile pour certains, difficile pour d'autres. Pour beaucoup de programmes qui respectaient les règles de la gestion de mémoire de Windows 2.x, la mise à jour vers Windows 3.0 n'a pas été un travail éprouvant. En gros, il fallait s'assurer que le programme tournait correctement en mode protégé et qu'il ne considérait pas la police système comme une police à espacement fixe.

Quand Windows 3.0 est arrivé, tout le monde savait qu'il lui manquait TrueType. TrueType est la technologie de contour des polices qui nous promet un véritable Wysiwyg sur le Macintosh et sous Windows. Comme nous le verrons, il faudra modifier quelque peu les programmes pour profiter pleinement des avantages de TrueType.

Cela mis à part, le passage de Windows 3.0 à la version 3.1 se trouvera nettement plus aisé que celui des versions 2.x à la version 3.0.

Certaines des nouvelles fonctions de Windows 3.1 sont attendues et désirées depuis Windows 1.0 — nous avons maintenant une API standard pour la boîte de dialogue Ouvrir un fichier.

D'autres fonctions semblent bizarres, l'absence des algorithmes de décompression Lempel-Ziv constituait-elle une grosse lacune?

Windows 3.1 représente également une plate-forme pour OLE, qui sera sans doute, depuis DDE, le plus fameux et le plus énervant des STL (pour Sigle en trois lettres) de Windows (OLE : «Object Linking and Embedding», liaison et incorporation d'objets; DDE : «Dynamic Data Exchange», échange dynamique de données). Comme le DDE, le support OLE représentera un élément courant de comparaison pour les bancs d'essai de logiciels sous Windows. Et il faudra l'intégrer à votre application, même si elle ne possède rien qui puisse être utilisé par l'OLE.

Le refus du mode réel

L'une des modifications les plus intéressantes de Windows 3.1 ne fonctionnera pas en mode réel. De plus en plus nombreuses sont les applications qui refusent de tourner en mode réel, et la majorité des utilisateurs de Windows possède le matériel nécessaire pour exécuter Windows en mode standard (un processeur 286 ou supérieur, avec au moins 1 Mo de mémoire).

Ne plus employer le mode réel, cela signifie que l'on peut écrire des programmes Windows 3.1 sans se préoccuper de verrouiller et de déverrouiller sans cesse les blocs de mémoire globale. On peut allouer un bloc, le verrouiller pour récupérer un pointeur et conserver ce der-

Dans la jeune industrie de l'informatique, un vieux dicton affirme qu'il ne faut jamais acheter ni développer pour un système d'exploitation dont le numéro de version se termine par «point zéro». Il vaut mieux attendre les versions «point un» ou «point zéro un», lesquelles se trouvent débarrassées des bogues de la précédente (ou du moins des plus flagrants).

nier jusqu'au moment où le bloc sera libéré. On peut ensuite obtenir le handle du bloc à l'aide de GlobalHandle, le déverrouiller et le libérer.

Et l'utilisateur?

Il fut un temps où les programmeurs Windows travaillaient uniquement avec la ligne de commande du DOS, en utilisant des applications DOS et en n'exécutant Windows que pour tester leurs programmes. Mais maintenant, le programmeur Windows est également devenu un utilisateur Windows, grâce à des systèmes de programmation comme Borland C++, Turbo Pascal pour Windows, Microsoft Visual Basic et le système de développement graphique de QuickC pour Windows.

Les utilisateurs s'apercevront que la version 3.1 a fait quelques progrès. Le plus significatif concerne le nouveau Gestionnaire de fichiers. Dans la version 3.0, celui-ci est une application MDI (Multiple Document Interface) qui affiche une fenêtre contenant une arborescence et éventuellement d'autres fenêtres présentant la liste des fichiers d'un sous-répertoire . La première version du Gestionnaire de fichiers pose un gros problème lorsque l'arborescence change de disque, en local ou en réseau. Après chaque changement, il doit reconstruire l'arborescence.

Avec Windows 3.1, chaque fenêtre fille du Gestionnaire se trouve séparée en deux. La partie gauche présente une arborescence et la droite fournit le contenu d'un répertoire particulier (*voir figure 1*).

Il est possible de créer de nouvelles fenêtres, même avec plusieurs disques. Le Gestionnaire de fichiers se souvient de toutes les fenêtres de répertoire ouvertes. Si l'on quitte Windows pour y revenir ensuite, le Gestionnaire de fichiers rouvre l'ensemble des fenêtres qui étaient ouvertes lorsqu'on l'a quitté. C'est appréciable si l'on doit sortir fréquemment de Windows.

Le Gestionnaire de fichiers de Windows 3.1 montre également de meilleures performances que celui de la version 3.0. Sur mon disque dur ayant 500 répertoires, il fallait trente-cinq secondes au Gestionnaire de Windows 3.0 pour construire l'arborescence complète. Celui de Windows 3.1 ne met que vingt-neuf secondes. Et j'ai entendu dire que l'amélioration est encore plus nette sur un réseau. Un nouveau groupe de programmes, intitulé «Startup», a été ajouté au Gestionnaire de programmes. Si l'on place un programme dans ce groupe, il est exécuté automatiquement lors du démarrage de Windows. C'est beaucoup plus pratique que l'emploi de l'option Run du fichier WIN.INI.

Windows 3.1 supporte également les économiseurs d'écran, ces programmes farfelus qui accomplissent de curieuses choses sur votre écran lorsqu'on n'a pas utilisé le clavier ou la souris durant un certain temps. Le protocole d'économie d'écran de Windows 3.1 est identique à celui de Windows avec les Multimedia Extensions.

Les erreurs inacceptables de l'API

L'une des meilleures choses de Windows en mode protégé est l'erreur «Application non récupérable» (UAE, pour Unrecoverable Application Error). Le mode protégé permet à Windows de piéger les applications fautives et (du moins l'espérons-nous) de les fermer avant qu'elles n'endommagent le reste du système.

L'une des pires choses, en ce qui concerne Windows en mode protégé, c'est aussi l'erreur «Application non récupérable». Elle ne donne pas la moindre information sur la cause du problème — le programme, une DLL, un appel de fonction incorrect, une fonction défectueuse? — ni aucun moyen d'y remédier. Même si le programme est terminé, cela ne veut pas dire que Windows a correctement nettoyé la mémoire, il peut demeurer dans un état instable ou encore se bloquer complètement.

L'UAE est devenue la «caractéristique» la plus détestée de Windows 3.0 et il faut absolument l'améliorer.

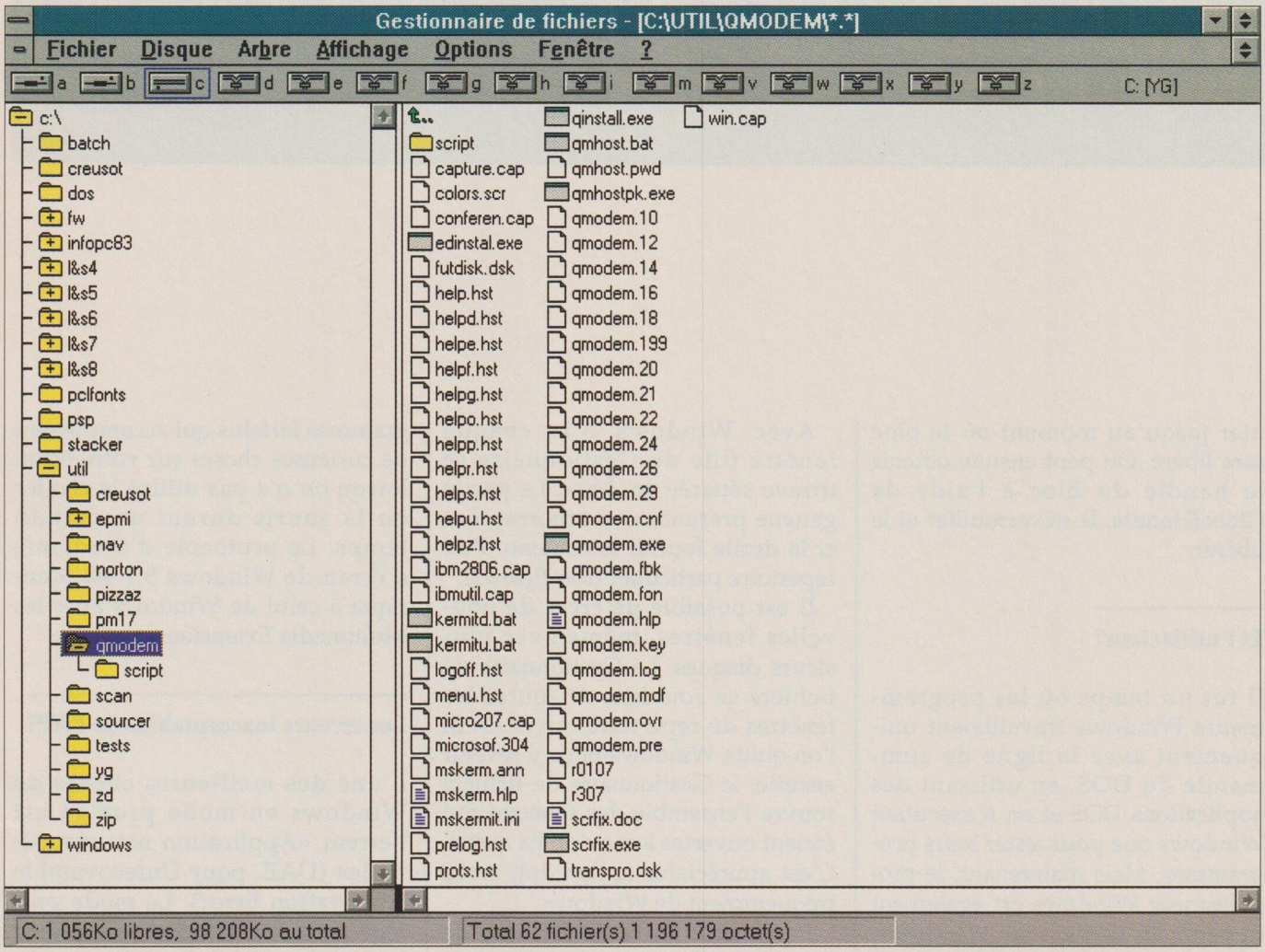


Figure 1 : le Gestionnaire de fichiers de Windows 3.1.

Bien entendu, c'est plus facile à dire qu'à faire. L'implémentation du mode protégé dans Windows (même dans la version 3.1) ne profite pas pleinement des avantages des processeurs 286 et 386 pour séparer les tâches. Toutes les applications Windows, et Windows lui-même, partagent la même table de descripteurs locale (LDT, pour Local Descriptor Table).

Paramètres incorrects

Les enquêtes de Microsoft montrent qu'environ 40% des UAE se produisent dans les applications elles-mêmes et sont des erreurs d'adressage, tandis que les 60% restant sont dus à des paramètres incorrects dans des appels à l'API.

Bien entendu, le terme «paramètre incorrect» peut être sujet à discussion. Mon plus récent problème, concernant ces mystérieuses UAE de Windows 3.0, était provoqué par un appel à la fonction PolyPolygon avec la valeur 1 dans le dernier paramètre. Ce bogue a été corrigé dans la version 3.1. Voyons les choses en face, les appels à l'API système devraient vérifier que les paramètres sont valides et bénéficier d'une protection maximale. Pour l'instant une erreur «Application non récupérable» est trop souvent une erreur «API non rectifiée».

Microsoft s'est efforcé de limiter autant que possible les UAE sous Windows 3.1, grâce à un meilleur contrôle des paramètres. La boîte de message UAE est également différente. Sous Windows 3.0 elle affichait simplement :

Erreur Application Non Récupérable.
Quitte l'application en cours.

Sous Windows 3.1, la boîte de message correspondante fournit quelques informations supplémentaires pour aider les développeurs à dépister la cause du problème. Elle affiche quelque chose de ce genre :

Erreur d'application BADAPP a provoqué une erreur de protection générale dans le module BADAPP.EXE à l'adresse 000100AE.

Pressez le bouton FERMER
pour fermer BADAPP.

Un petit programme très pratique est procuré avec Windows 3.10. Il s'intitule Dr.Watson et présente davantage d'informations lorsqu'une UAE se produit. Dr.Watson crée un fichier journal contenant l'adresse de l'erreur, les instructions désassemblées qui ont entraîné l'erreur, le contenu des registres du CPU, le contenu de la pile, une liste des tâches et des DLL chargées à ce moment sous Windows, et quelques autres informations internes concernant l'état de l'environnement. Grâce à Dr.Watson, j'ai pu découvrir en seulement quelques heures la cause d'une UAE dans un programme qui me tracassait depuis plusieurs mois.

L'évolution du Wysiwyg

La définition des traitements de texte Wysiwyg a changé au cours des dernières années. Il fut un temps où WordStar 3.x était considéré Wysiwyg simplement parce qu'il affichait correctement les ruptures de ligne et de page! La plupart des traitements de texte ne faisaient pas cela à cette époque. Plus tard, Microsoft Word a été appelé à son tour Wysiwyg parce qu'il montrait les caractères soulignés, italiques et gras. (Bien que toutes les polices eussent la même taille!) Puis ce fut le tour de Windows Write, parce qu'il pouvait afficher plusieurs types et plusieurs tailles de polices.

Mais, malgré l'intégration de différentes polices de tailles diverses dans Windows, l'affichage vidéo de documents formatés ne correspondait jamais parfaitement à ce qui était imprimé. Le problème venait du fait que les polices d'affichage Windows étaient stockées sous forme de bitmap. Ces polices bitmap (ou matricielles) se trouvaient limitées par la résolution de l'affichage et par leur propre taille, ce qui rendait leur manipulation difficile. Les polices matricielles pouvaient être élargies en multipliant par un entier les rangées et les colonnes de pixels, mais cela donnait aux contours un désagréable aspect d'escalier. Les seules concurrentes des polices matricielles étaient les polices de type «stroke font» (polices vectorielles). Comme elles étaient dessinées à l'aide de segments de droites, ces polices pouvaient être élargies ou réduites. Cela peut convenir à un traceur mais ne présente aucun intérêt pour la typographie.

Entre temps, les imprimantes ont fait des progrès. PostScript, en particulier, a démontré la puissance des polices «outline», indéfiniment transformables — des polices définies par des contours (outlines), dont on peut modifier la taille, et qui peuvent être inclinées ou subir une rotation.

Les programmeurs travaillant avec l'interface de programmation graphique d'OS/2 1.1 (GPI, pour Graphics Programming Interface) ont pu observer la souplesse des polices outline. Celles de la GPI présentaient néanmoins une grosse lacune. Le redimensionnement simple n'effectuait aucun ajustement pour respecter la typographie. Par exemple, comme certaines valeurs sont arrondies, les deux pattes d'un H en Helvetica pouvaient avoir un pixel de différence en épaisseur. Sur l'écran, avec des caractères de taille normale, cette erreur produisait un texte pratiquement illisible.

Le standard TrueType

La correction des contours d'une police afin de respecter l'esthétique typographique est appelée «hinting» (une technique spécifique de lissage). Windows n'a pas proposé de polices outline «lissées» jusqu'à ce que d'autres compagnies éditent des utilitaires spécifiques, comme Adobe Type Manager.

Windows 3.1 adopte une nouvelle technologie de polices outline intitulée «TrueType», laquelle fut développée initialement par Apple pour le Macintosh (implémentée dans le système d'exploitation Système 7.0)

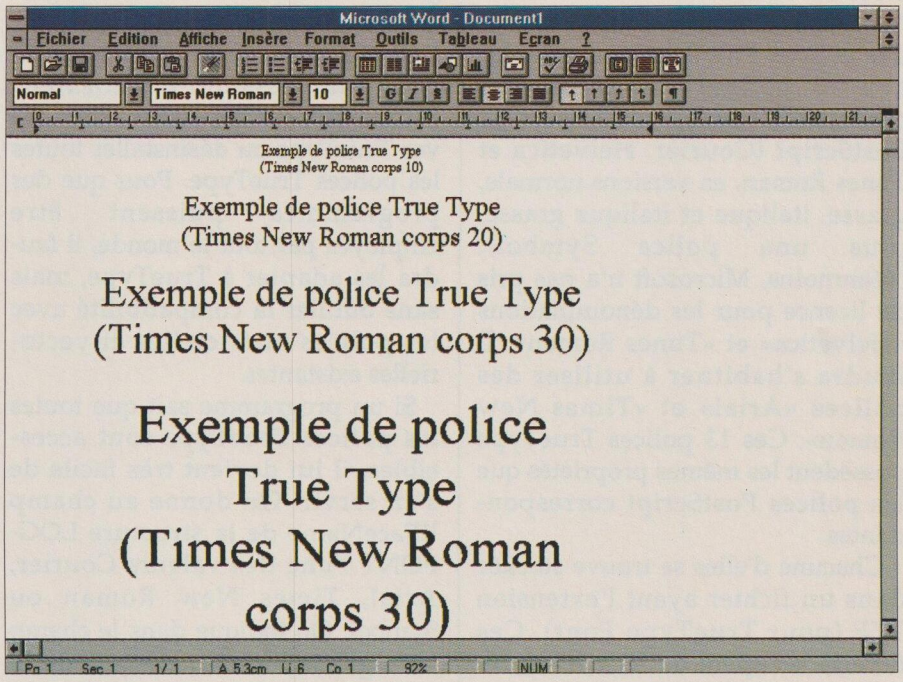


Figure 2 : un jeu de caractères TrueType.

Figure 3 : le jeu de caractères Symbol de TrueType.

avant d'être achetée sous licence par Microsoft. En tant que technologie outline incluse dans les deux plates-formes graphiques les plus répandues, TrueType se place en bonne position pour devenir un standard de l'industrie micro-informatique. Et il s'agit d'un standard ouvert — tout fabricant de polices ayant la possibilité de développer des polices au format TrueType. Les polices TrueType peuvent être employées par tous les périphériques de sortie graphique supportés par Windows.

Microsoft fournit avec Windows 3.1 des polices TrueType équivalentes aux 13 polices standards que connaissent bien les utilisateurs de PostScript (Courier, Helvetica et Times Roman, en versions normale, grasse, italique et italique grasse, plus une police Symbol). Néanmoins, Microsoft n'a pas pris de licence pour les dénominations «Helvetica» et «Times Roman». Il faudra s'habituer à utiliser des polices «Arial» et «Times New Roman». Ces 13 polices TrueType possèdent les mêmes propriétés que les polices PostScript correspondantes.

Chacune d'elles se trouve stockée dans un fichier ayant l'extension TTF (pour TrueType Font). Ces fichiers occupent de 53 à 70 Ko, et l'on peut supposer que les polices TrueType des autres fabricants auront à peu près la même taille. Bien qu'ils soient plus grands que les fichiers des polices matricielles, n'oubliez pas qu'un seul fichier TrueType fournit une police disponible pour tous les périphériques de sortie et toutes les résolutions d'affichage, et dans toutes les tailles.

Compatibilités

Microsoft recommande que les programmes Windows se servent uniquement des polices TrueType pour l'affichage du texte sur l'écran et pour l'impression. L'utilisateur Windows peut cependant désactiver TrueType ou désinstaller toutes les polices TrueType. Pour que des programmes puissent être employés par tout le monde, il faudra les adapter à TrueType, mais sans oublier la compatibilité avec les polices matricielles et vectorielles existantes.

Si un programme sait que toutes les polices TrueType sont accessibles, il lui devient très facile de s'en servir. On donne au champ lfFaceName de la structure LOGFONT l'une des valeurs Courier, Arial, Times New Roman ou Symbol. On indique dans le champ lfHeight la hauteur désirée de la police, en unités logiques. (Normalement, ce champ spécifie une hauteur incluant la «tête» des caractères, correspondant à l'interligne souhaitée. On utilise une valeur négative si l'on veut que lfHeight représente le point de la police.) On donne éventuellement la valeur 700 (pour le gras) à lfWeight et la valeur 1 aux champs lfItalic, lfUnderline ou lfStrikeOut. Il est également possible de donner une valeur différente de 0 à lfWidth si l'on ne désire pas une largeur standard. Tous les autres champs de la structure LOGFONT peuvent être mis à 0. Ensuite, on passe à CreateFontIndirect un pointeur sur la structure LOGFONT, et nous voilà prêt à sélectionner le handle de la police dans un contexte de périphérique.

Lorsque l'on emploie les polices TrueType sur l'écran, les contours sont dimensionnés à la hauteur souhaitée, puis convertis en bitmap. Une fois que ces bitmap sont stockés en mémoire, leur affichage se révèle aussi rapide que celui des polices matricielles.

Lorsque l'on se sert de TrueType avec une imprimante, la GDI peut soit envoyer les polices à l'imprimante sous la forme de bitmaps ou de polygones pleins, soit les passer au pilote d'imprimante qui convertit les données des polices TrueType en polices téléchargeables. Les

15 ans d'exploration

ou l'art de parfaire une carte

1492 / 1992 une carte est toujours un instrument de précision

Parcours ARN

ARN sillonne les voies de transmission depuis 15 ans. Son sens de pionnier le guide toujours dans la recherche de nouveaux horizons de connexion à distance. Navigateurs expérimentés, ses ingénieurs savent gréer tout type de logiciel et sont aussi experts dans le tracé de nouvelles cartes. Depuis 1977, cette société française conseille, conçoit des architectures et des solutions avec un sens aigu de l'orientation. ARN : l'assurance de suivre la bonne voie.

Longitude ARN3270

Le logiciel ARN3270, entouré des autres émulateurs ARN, apparaît comme un archipel de simplicité et de fiabilité dans l'océan du monde synchrone. Evitant le récif des configurations multiples et complexes, vous y trouverez le fruit d'un parcours sans dérive. ARN3270 : l'émulation hissée au rang de perfection.

Latitude ARNDX

Véritable découverte, la carte ARNDX marque son temps du sceau de l'universalité. Première carte permettant à des note books de franchir le cap des 9600 bps et d'atteindre les 14400 bps, elle intègre huit modems (V21, V22, V22bis, V23, V27ter, V29, V32, V32bis) et propose un fax en standard. Avec ARNDX, l'idée de connexion n'a plus de limite.

La carte ARNDX permet à tous les logiciels du monde de monter à bord ! Un équipage peut en remplacer un autre, mais c'est toujours l'équipage d'origine qui vous assure le voyage le plus rapide et le plus sûr.

A . R . N.

L'ART DE COMMUNIQUER

51 - 63 rue Gaston Lauriau
93100 Montreuil
(1) 48 57 52 00

Service lecteurs nº 23 (dernière page)

polices italiques TrueType sont vraiment en italique, et pas simplement des versions inclinées des polices normales. On le remarque bien avec les caractères «a» et «f» des polices Times New Roman et Times New Roman Italic.

Certains des codes ASCII inutilisés, entre 80H et 9FH, sont désormais employés comme symboles pour la PAO (*voir figure 2*). La police Symbol de TrueType apparaît compatible avec celle de PostScript (*voir figure 3*).

Les applications Windows existantes qui emploient plusieurs polices appellent généralement EnumFonts pour que l'utilisateur puisse faire son choix. Certaines de ces applications pourront fonctionner sans modification avec TrueType, d'autres n'en auront pas la possibilité. En simplifiant, cela tient au fait que ces dernières excluent délibérément les polices vectorielles en testant le bit 0 du paramètre nFontType de la fonction callback associée à EnumFonts. Les applications qui écartent les polices vectorielles interdiront également les polices TrueType.

Comme les polices matricielles utilisent le jeu de caractères Ansi, et que les polices vectorielles emploient le jeu de caractères OEM (c'est-à-dire celui de l'IBM PC), certains programmes peuvent croire que seules les polices se servant du jeu OEM sont redimensionnables. C'est faux. Les programmes qui supposent cela devront être rectifiés pour devenir compatibles avec TrueType.

Identification

Une application Windows proposant un choix de polices vectorielles à l'utilisateur est en bonne position. Il lui sera très certainement possible d'employer les polices TrueType sans avoir besoin d'être modifiée.

Un programme peut identifier une police TrueType grâce au nouvel indicateur SCALABLE_FONTTYPE, dans le paramètre nFontType de la fonction callback associée à EnumFonts. Si l'indicateur est à 1, il s'agit d'une TrueType. Une autre méthode d'identification des polices TrueType consiste à appeler GetTextMetrics. Les quatre bits de poids faible du champ lfPitchAndFamily sont codés. Le bit 2 est le nouvel indicateur de TrueType.

L'usage exclusif de TrueType pour l'écran et pour l'imprimante revient simplement à ignorer les polices dont l'indicateur SCALABLE_FONTTYPE n'est pas validé (mis à 1). Et l'on résout quelques problèmes en n'utilisant que TrueType. D'abord, les polices auront le même aspect et la même taille sur l'écran et sur le papier. Ensuite, le même jeu de caractères sera disponible pour l'écran et pour l'imprimante. Enfin, nous avançons vers une compatibilité avec des documents TrueType créés sur Macintosh.

Malgré cela, les utilisateurs ayant investi dans l'achat de cartouches de polices et de polices téléchargeables pourraient être contrariés

Figure 4 : Windows 3.1 intègre un programme d'enregistrement de sons.

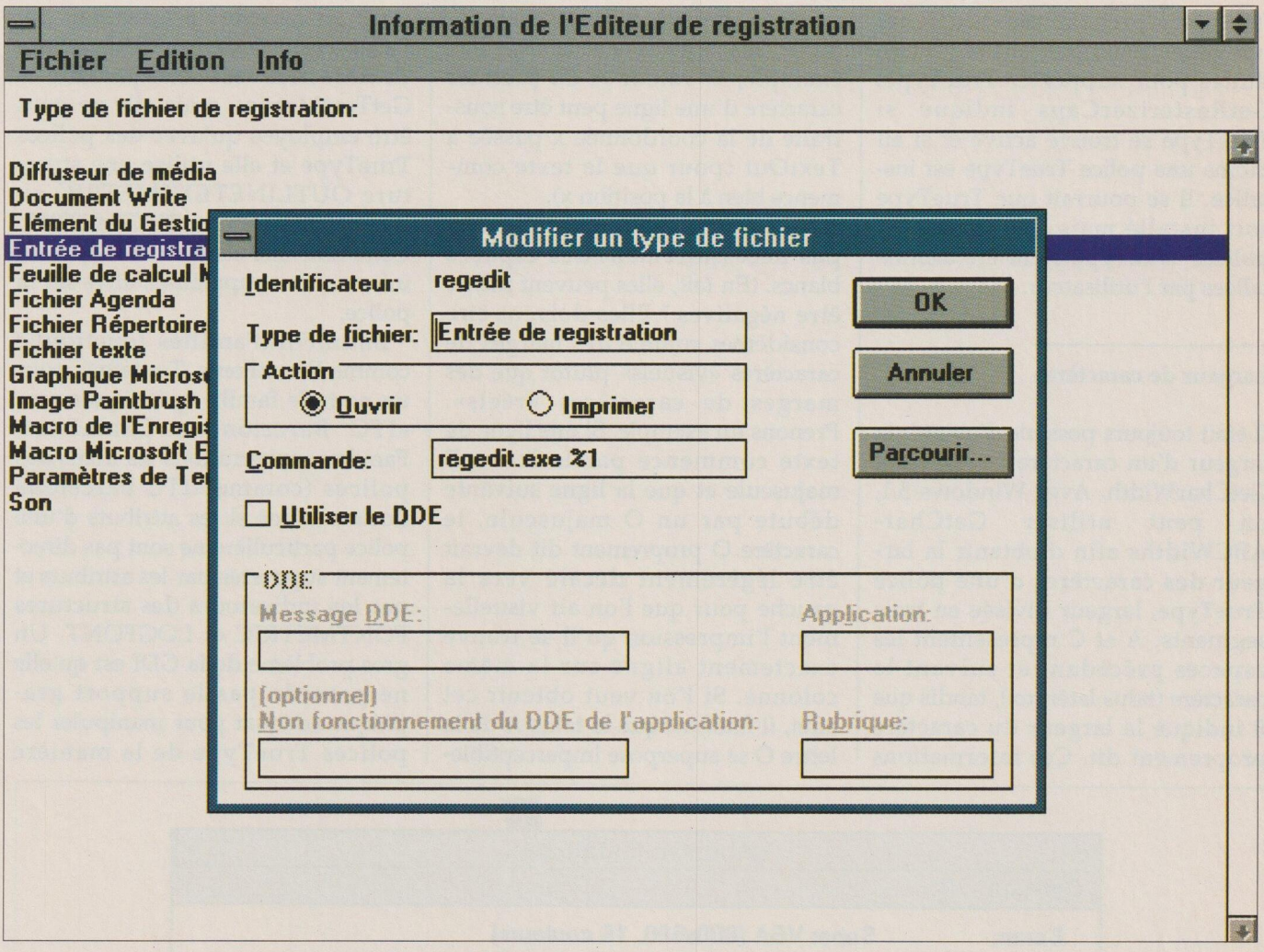


Figure 5 : Regedit définit le type de relations OLE d'une application.

de voir qu'il leur est interdit de s'en servir avec une application. Sans ignorer le cas de ceux susceptibles de désactiver TrueType ou de désinstaller les polices TrueType. Une meilleure approche consiste à proposer toutes les polices d'impression aux utilisateurs, tout en les encourageant à utiliser TrueType. Si la fonction EnumFonts est employée pour placer la liste des polices dans une boîte liste, les noms des polices TrueType apparaissent avec des caractères en minuscules et en majuscules, alors que tous les autres noms de polices se trouvent affichés entièrement en majuscules. C'est la troisième façon d'identifier une police TrueType. Visuellement, l'utilisateur pourra faire une distinction entre les différents types de polices. Vous voudrez probablement créer une boîte de dialogue présentant les noms des polices et les tailles disponibles, pour que l'utilisateur puisse faire son choix et définir éventuellement la taille d'une police redimensionnable.

L'un des principaux intérêts des polices outline réside dans la possibilité de manipuler les contours comme n'importe quel autre objets graphiques. Par exemple, il est très facile de faire tourner les caractères. Néanmoins, en implémentant TrueType dans Windows 3.1, Microsoft s'est davantage concentré sur les performances et la lisibilité que sur les effets spéciaux.

Dans la première version bêta de Windows 3.1, les polices TrueType ne répondaient pas aux directives des champs lfEscapement ou lfOrientation de la structure LOGFONT.

Windows 3.1 contient toujours les anciennes polices matricielles et vectorielles, mais les premières ont changé de nom. L'ancienne police Helv est maintenant appelée MS Sans Serif, et la Tms Rmn est devenue MS Serif. Si votre programme se réfère à ces polices par leurs noms antérieurs, dans des appels à CreateFont ou à CreateFontIndirect, ne vous en souciez pas, cela fonctionnera. Mais si vous appelez EnumFonts pour rechercher les anciens noms, il vous faudra modi-

fier votre programme. Plusieurs nouvelles fonctions ont été introduites pour supporter TrueType. GetRasterizerCaps indique si TrueType se trouve activé et si au moins une police TrueType est installée. Il se pourrait que TrueType soit installé mais que toutes les polices TrueType aient été désinstallées par l'utilisateur.

Largeur de caractères

Il était toujours possible d'obtenir la largeur d'un caractère en appelant GetCharWidth. Avec Windows 3.1, on peut utiliser GetCharABCWidths afin d'obtenir la largeur des caractères d'une police TrueType, largeur divisée en trois segments. A et C représentent les espaces précèdant et suivant le caractère (talus latéraux), tandis que B indique la largeur du caractère proprement dit. Ces informations peuvent servir à orienter correctement les lignes de texte. Par exemple, la valeur A du premier caractère d'une ligne peut être soustraite de la coordonnée x passée à TextOut (pour que le texte commence bien à la position x).

Les valeurs A et C ne représentent pas nécessairement des espaces blancs. (En fait, elles peuvent même être négatives.) Elles doivent être considérées comme des marges de caractères «visuels» plutôt que des marges de caractères «réels». Prenons un exemple. Si une ligne de texte commence par la lettre E majuscule et que la ligne suivante débute par un O majuscule, le caractère O proprement dit devrait être légèrement décalé vers la gauche pour que l'on ait visuellement l'impression qu'il se trouve exactement aligné sur la même colonne. Si l'on veut obtenir cet effet, il faudrait que le talus A de la lettre O se superpose imperceptiblement au caractère proprement dit. La nouvelle fonction GetOutlineTextMetrics est comparable à GetTextMetrics, mais elle ne peut être employée qu'avec des polices TrueType et elle utilise une structure OUTLINETEXTMETRIC au lieu d'une structure TEXTMETRIC. Cette nouvelle structure fournit des informations supplémentaires sur la police.

EnumFontFamilies fonctionne comme EnumFonts. Toutefois, pour un nom de famille spécifié (comme «ITC Barcelona»), EnumFontFamilies renverra tous les noms des polices (comme «ITC Barcelona Book») même si les attributs d'une police particulière ne sont pas directement supportés par les attributs et par les indicateurs des structures FONTMETRIC et LOGFONT. Un gros problème de la GDI est qu'elle ne possède pas le support graphique adéquat pour manipuler les polices TrueType de la manière

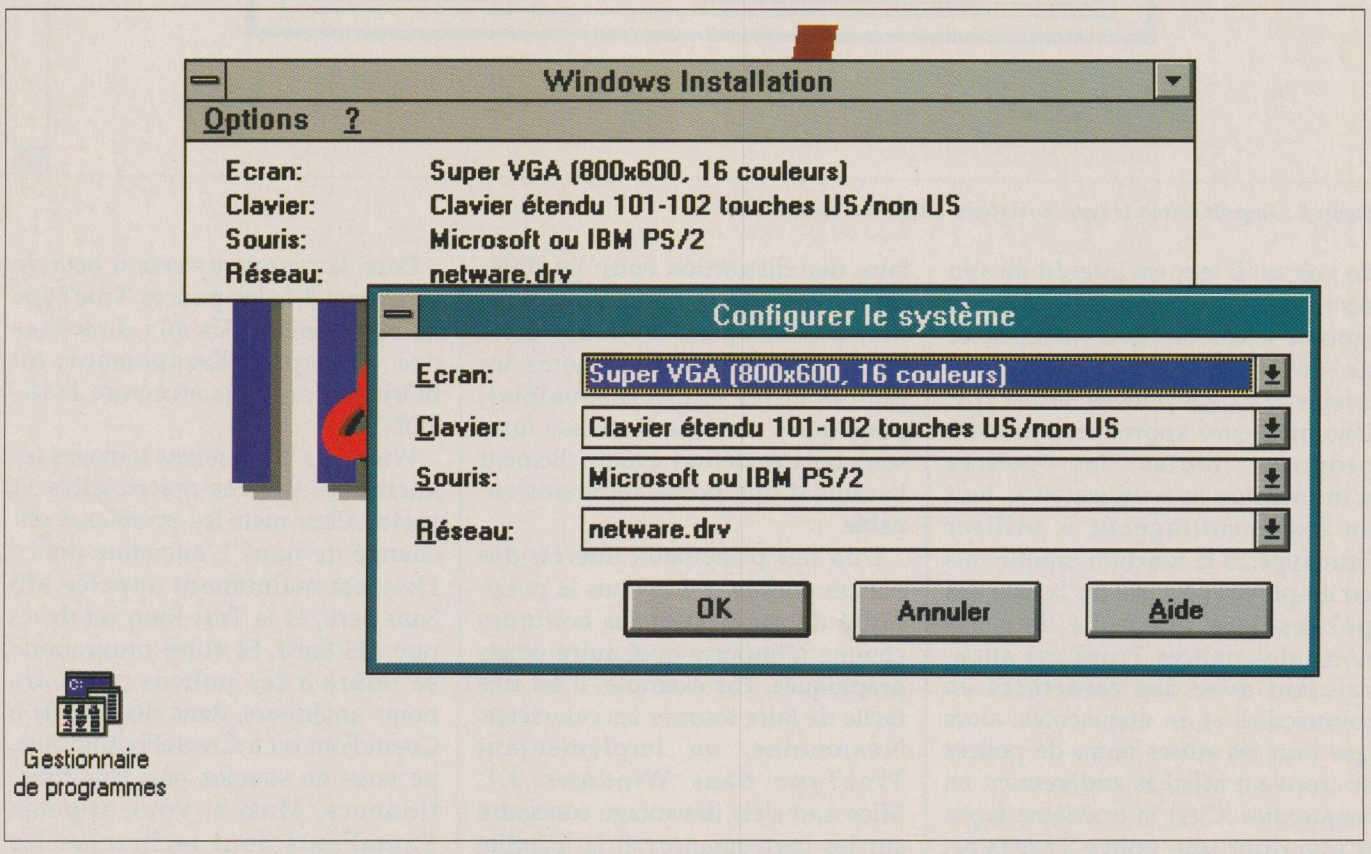


Figure 6 : Windows 3.1 gère d'origine la résolution XGA, S-VGA, et Tiga.

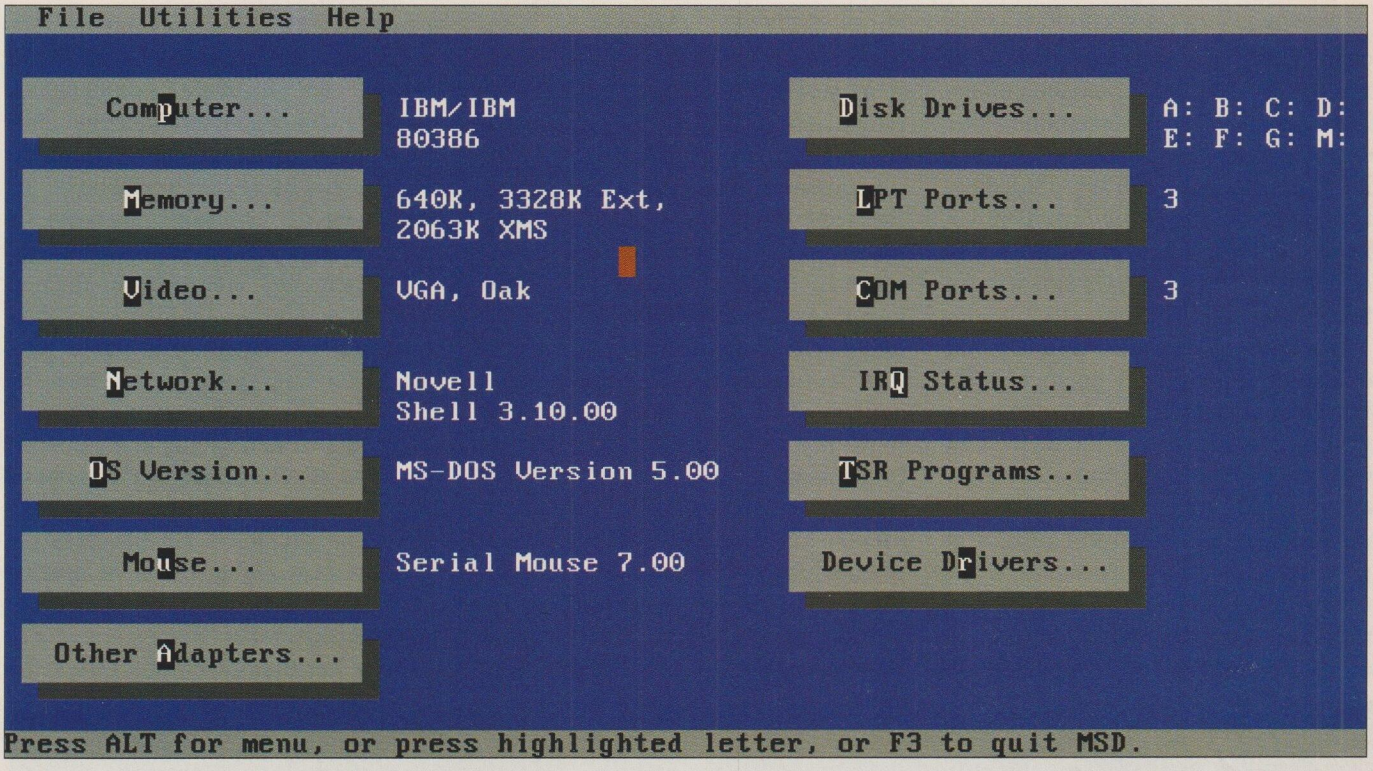


Figure 7 : Windows 3.1 est livré avec un outil de diagnostic qui s'exécute en mode texte.

dont les polices outline sont manipulées par PostScript ou par la GPI. Il lui manque en particulier le support de chemin et les transformations de matrices, qui devraient être inclus dans l'API de Windows 32 bits.

En compensation, Microsoft a introduit GetGlyphOutline. Cette fonction permet d'obtenir la courbe de contour d'un caractère. Le contour peut être lissé ou non. Quant au caractère, il peut être retourné soit comme une bitmap, soit comme un polygone, et l'on peut encore spécifier une matrice de rotation. Cela demande du travail, et ne s'avère pas vraiment pratique, mais, en cas de besoin, la fonction est là.

Pour terminer sur ce sujet, disons qu'il existe une nouvelle fonction qui permet d'employer à la fois les anciennes polices et les polices TrueType. Il s'agit de GetTextExtentEx, une forme étendue de la fonction GetTextExtent. Elle reçoit comme paramètres une chaîne et une distance et indique le nombre de caractères de la chaîne qui pourront tenir dans cette distance.

Un DDE simplifié?

Pendant plusieurs années, le DDE a été l'un des domaines les plus... stimulants de la programmation Windows. Le DDE emploie le système normal d'échange de messages de Windows pour implémenter la communication entre processus.

Personne ne nie la puissance du DDE pour permettre aux applications Windows d'échanger des informations, mais il peut se révéler très difficile de travailler avec les messages DDE (et avec les diverses règles de gestion de la mémoire et des atomes).

Windows 3.1 propose une solution partielle à ces problèmes avec la DDEML (pour Dynamic Data Exchange Management Library, bibliothèque de gestion de l'échange dynamique d'informations). Il s'agit d'une couche de l'API, à base de fonctions, construite sur l'ancien DDE à base, lui, de messages. Bien entendu, l'ancien protocole DDE reste toujours disponible, et les programmes qui l'emploient pourront continuer de communiquer avec les applications qui se servent de la DDEML. Notons que cette dernière n'est utilisable qu'en mode protégé. Bien que le but de la DDEML soit de simplifier et de rendre plus sûre la programmation du DDE, elle peut décourager le programmeur qui vient juste de se familiariser avec la complexité du protocole de messages DDE. La DDEML comporte 26 nouvelles fonctions, 9 nouvelles structures et 142 nouvelles constantes, dont 16 sont des types de transactions destinés à une fonction callback.

La terminologie de la DDEML est proche de celle du DDE. Il se produit toujours une conversation DDE entre un serveur (le programme qui

possède les informations) et un client (le programme qui veut obtenir ces dernières). Il existe cependant quelques différences. Par exemple, les conversations sous DDE emploient un nom d'application, un nom de topic (sujet) et un nom d'item (élément). Avec DDEML, nous avons un service, un topic et un nom d'item.

Nous allons examiner maintenant quelques fonctions et protocoles de la nouvelle DDEML. Tout programme qui utilise la DDEML (soit comme client, soit comme serveur) appelle DdeInitialize très tôt pour enregistrer une fonction callback et pour spécifier d'éventuels filtres de transaction. DdeInitialize renvoie un identificateur d'instance employé par toutes les autres fonctions DDEML. Quand le programme en a fini avec la DDEML, il appelle DdeUninitialize. La fonction callback DDE a huit paramètres, le premier est le type de transaction. Comme une procédure de fenêtre, la fonction callback DDE se sert d'une importante instruction d'aiguillage pour gérer les transactions. La plupart des autres paramètres sont relatifs à la transaction. Comme dans le cas du DDE, le service, le topic et le nom d'item sont des chaînes de texte. Cependant, celles-ci chaînes ne sont pas passées directement aux fonctions DDEML. Au lieu d'employer des atomes globaux pour représenter les chaînes, un programme DDEML utilise cinq nouvelles fonctions qui travaillent avec des handles de chaîne. La fonction DdeCreateStringHandle reçoit une chaîne et renvoie un handle de chaîne, tandis que DdeQueryString reçoit un handle de chaîne et renvoie d'abord la longueur de la chaîne, ensuite la chaîne proprement dite.

Comme les atomes globaux, toutes les chaînes DDEML se trouvent stockées dans une table globale. (En fait, pour les chaînes, la DDEML se sert des atomes globaux de façon interne.) Les chaînes ne sont pas répétées dans cette table et chacune s'associe à un compteur d'utilisation. Mais les handles de chaîne sont uniques — la même chaîne peut être référencée par deux handles différents. Pour déterminer si deux handles de chaîne font réfé-

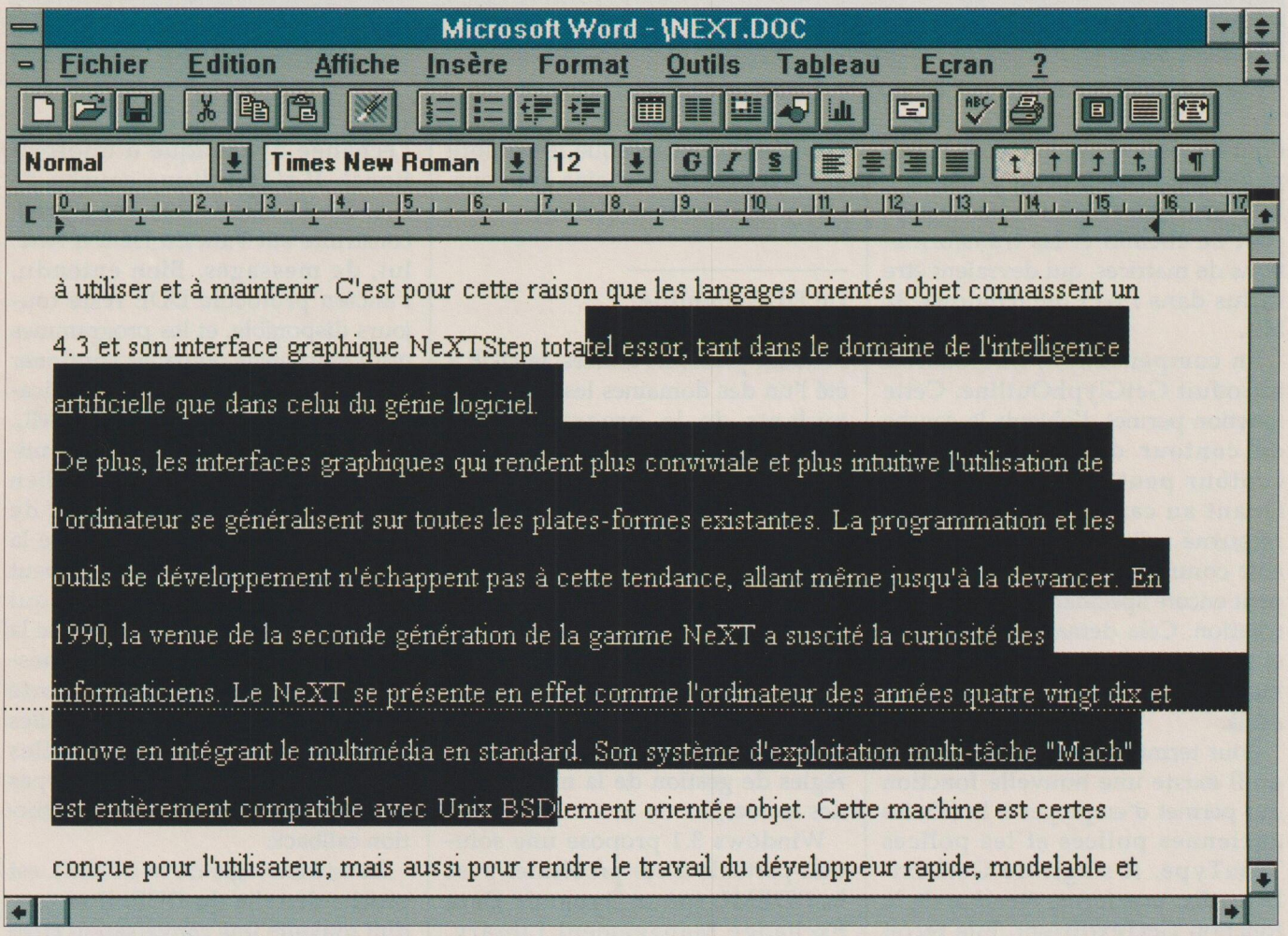


Figure 8 : l'un des principaux apports de Windows 3.1 est la fonction Drag and Drop qui consiste à faire glisser une zone sélectionnée.

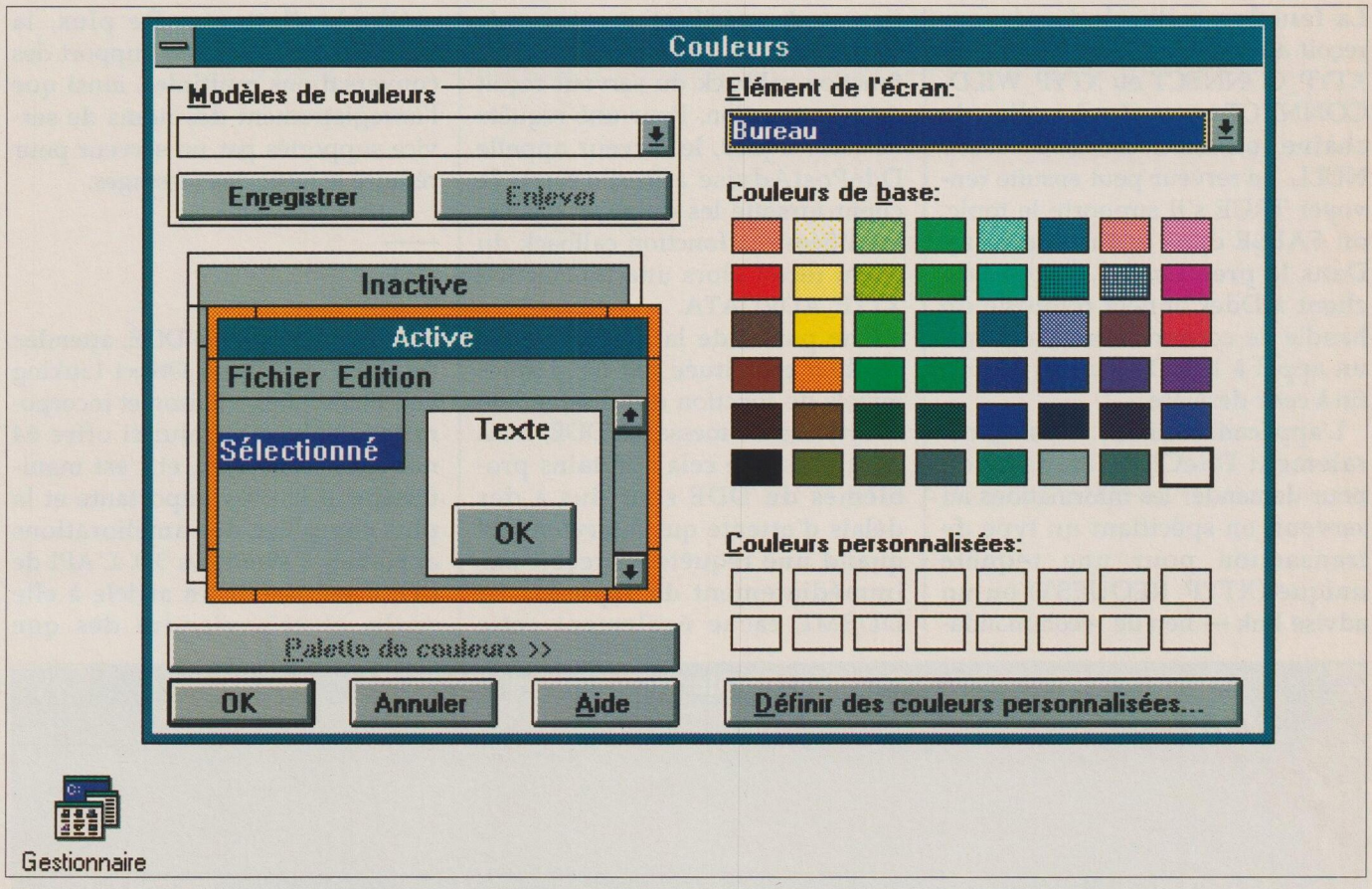


Figure 9 : sous Windows 3.1, l'utilisateur peut davantage personnaliser l'interface.

rence à la même chaîne, le programme appelle DdeCmpStringHandles.

Quand la DDEML passe un handle de chaîne à une fonction callback DDE, il est invalidé au retour de la fonction callback. Pour conserver le handle de chaîne, un programme peut employer DdeKeepStringHandle.

Règles d'utilisation

Les règles d'utilisation des handles de chaîne s'avèrent un peu plus simples et plus solides que celles des atomes globaux dans le DDE. DdeCreateStringHandle et DdeKeepStringHandle incrémentent le compteur d'utilisation, lequel est décrémenté par DdeFreeStringHandle.

Lorsque le compteur d'utilisation tombe à zéro, la chaîne est retirée de la table. Les appels d'incrémentation et de décrémentation doivent, bien sûr, s'équilibrer dans chaque application DDEML.

La DDEML implémente également six nouvelles fonctions de gestion des données en plus des fonctions standards Windows relatives à la mémoire globale. DdeCreateDataHandle accepte un pointeur sur des données, alloue un bloc de mémoire globale, y copie les données et renvoie un handle de données pour ce bloc. Lorsque ce handle est passé à une fonction callback, un programme peut copier les données dans la mémoire locale à l'aide de DdeGetData.

Le programme récepteur peut également accéder aux données (sans les copier) par des appels joints à DdeAccessData et à DdeUnaccessData. Quand le programme qui a créé le bloc de données n'en a plus besoin, il appelle DdeFreeDataHandle. La seule autre fonction de gestion de données est DdeAddData, qui peut ajouter des informations à un bloc de données, après qu'il ait été créé mais avant qu'il soit passé à une fonction DDEML.

Les fonctions et les règles DDEML de gestion des chaînes et de la mémoire sont plus simples et plus rationnelles que le code correspondant requis pour traiter les messages DDE.

Conversation

Une application cliente démarre une conversation DDE en appelant DdeConnect avec son handle d'instance et les handles de chaîne des noms de service et de topic (qui peuvent avoir pour valeur NULL).

La fonction callback du serveur reçoit alors un type de transaction XTYP_CONNECT ou XTYP_WILD-CONNECT si un des handles de chaîne (ou les deux) a la valeur NULL. Le serveur peut ensuite renvoyer TRUE s'il supporte le topic, ou FALSE dans le cas contraire. Dans le premier cas, l'appel du client à DdeConnect retourne un handle de conversation. Plus tard, un appel à DdeDisconnect mettra fin à cette dernière.

L'application cliente utilise généralement DdeClientTransaction pour demander les informations au serveur, en spécifiant un type de transaction pour une requête unique (XTYP_REQUEST) ou un advise link — lien de recommandation ou de spécification — pour la mise à jour (XTYP_ADVSTART). La fonction callback du serveur reçoit cette transaction. Pour une requête de mise à jour, le serveur appelle DdePostAdvise afin d'avertir le client lorsque les données ont été modifiées. La fonction callback du client reçoit alors une transaction XTYP_ADVDATA.

Une partie de la DDEML peut sembler constituée par de simples appels de fonction qui dissimulent le système de messages DDE, mais c'est plus que cela : certains problèmes du DDE sont dus à des délais d'attente qui interviennent quand une requête ne reçoit pas immédiatement de réponse. La DDEML cache également cette méthode d'attente. De plus, la DDEML implémente le support des conversations multiples, ainsi que l'enregistrement des noms de service supportés par un serveur pour réduire le trafic des messages.

OLE

Si vous avez aimé le DDE, attendez de voir l'OLE (pour Object Linking and Embedding : liaison et incorporation d'objets). Celui-ci offre 64 nouvelles fonctions, et c'est manifestement la plus importante et la plus complexe des améliorations apportées à Windows 3.0. L'API de l'OLE mériterait un article à elle seule, et cela viendra dès que

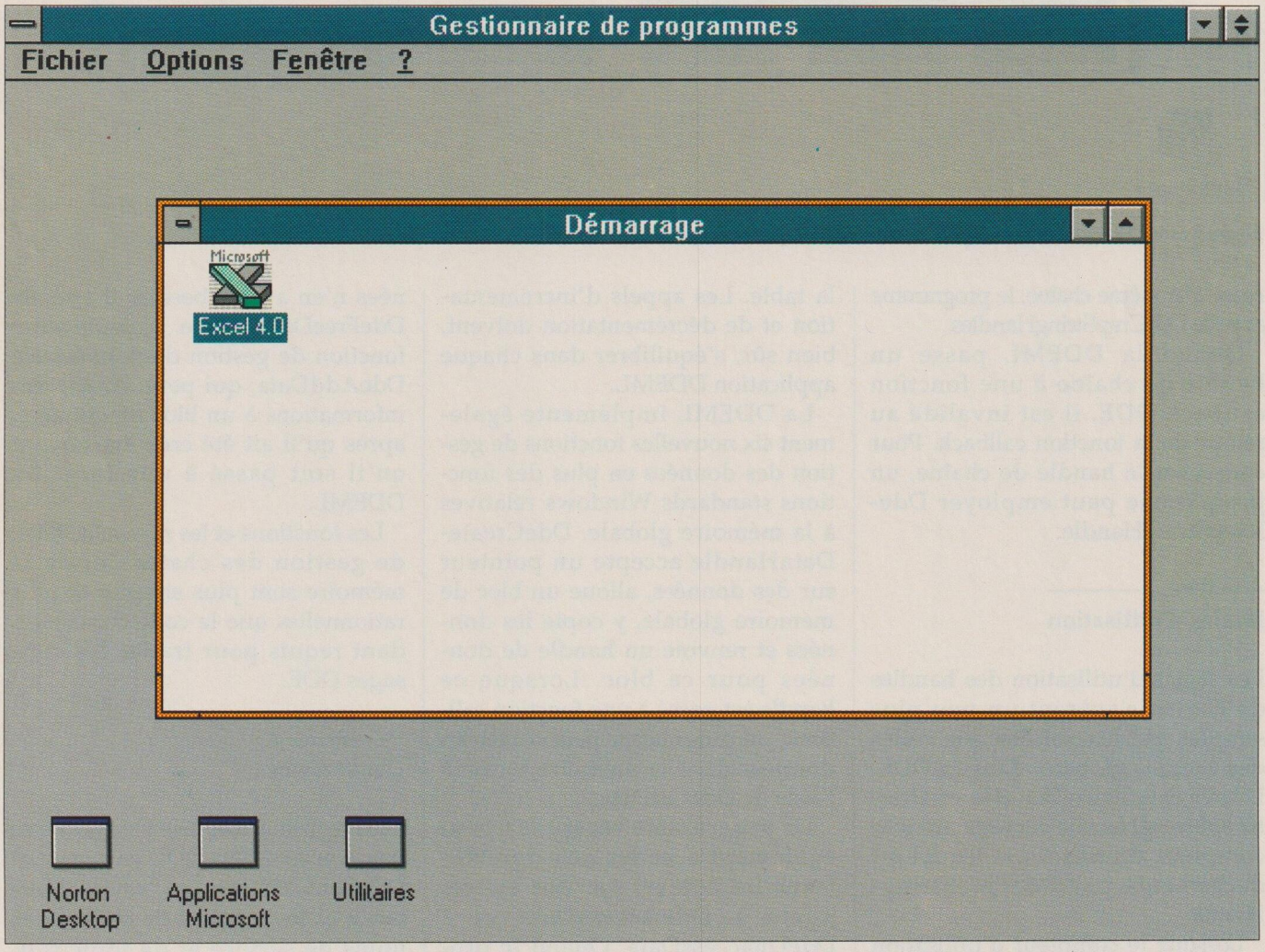


Figure 10 : le groupe de programmes Démarrage permet de définir la liste des applications qui doivent être chargées automatiquement.

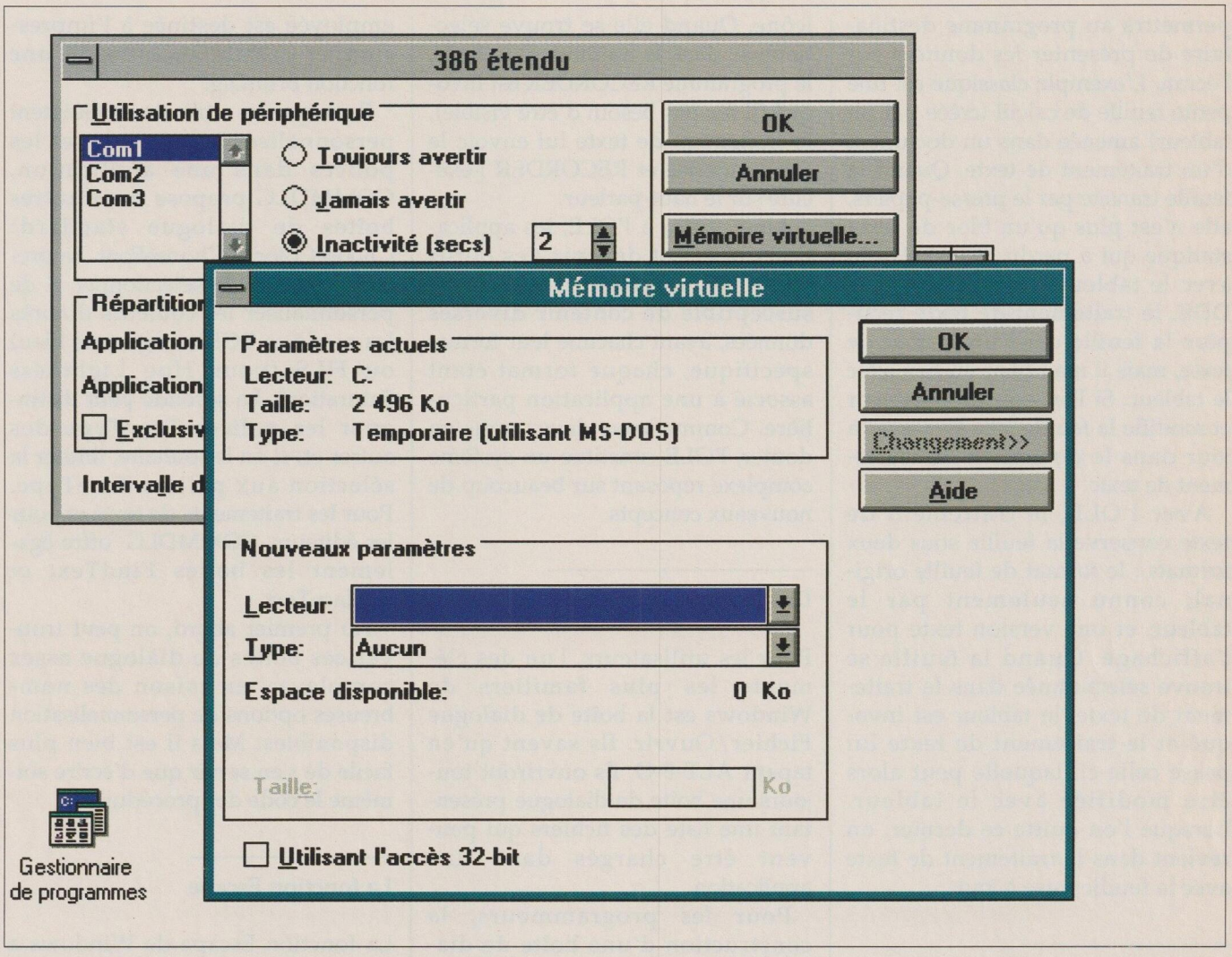


Figure 11 : en mode 386, il est possible d'effectuer des accès 32 bits sur le disque dur.

quelqu'un aura réussi à décrypter la documentation et à comprendre le mode d'emploi.

L'OLE déplace le focus utilisateur des applications vers les documents en définissant un format pour les documents composites. Ceux-ci sont capables de contenir plusieurs sortes de données, qui peuvent être reconnues et gérées par plusieurs applications. Cela permet à ces dernières de se concentrer sur les tâches qu'elles accomplissent le mieux, et l'utilisateur final peut alors exploiter une combinaison de ces applications pour construire un document particulier. On comprend mieux l'OLE en le comparant aux deux autres méthodes employées par les applications Windows pour partager des informations.

Données indépendantes

Quand il utilise le Presse-papiers, un programme obtient des données en provenance d'un autre programme dans un format standard, généralement du texte ASCII, une image bitmap ou un métafichier. Ces données sont indépendantes — il n'y a pas de lien avec le programme qui les a placées dans le presse-papiers.

Lorsqu'il emploie le DDE, un programme recueille également des données venant d'un autre programme et, là encore, ces données ont un format standard (texte, bitmap ou métafichier). Cependant, le programme client peut maintenir un lien avec le programme serveur qui lui a fourni les informations. Si l'on modifie les données originales dans le programme serveur, elles peuvent être également mises à jour par le programme client.

L'OLE, lui aussi, permet à un programme de recevoir des informations à partir d'un autre programme. Mais ces données peuvent être procurées sous deux formats : soit un format qui n'est connu que par l'envoyeur, soit un format d'affichage (en général du texte, une bitmap ou un métafichier) qui

permettra au programme destinataire de présenter les données sur l'écran. L'exemple classique est une petite feuille de calcul (créée par un tableur) amenée dans un document d'un traitement de texte. Quand la feuille transite par le presse-papiers, elle n'est plus qu'un bloc de texte statique qui a perdu toute relation avec le tableur. En employant le DDE, le traitement de texte récupère la feuille dans un format de texte, mais il maintient un lien avec le tableur. Si l'on exécute le tableur et modifie la feuille, elle sera mise à jour dans le document du traitement de texte.

Avec l'OLE, le traitement de texte conserve la feuille sous deux formats : le format de feuille original, connu seulement par le tableur, et une version texte pour l'affichage. Quand la feuille se trouve sélectionnée dans le traitement de texte, le tableur est invoqué et le traitement de texte lui passe celle-ci, laquelle peut alors être modifiée avec le tableur. Lorsque l'on quitte ce dernier, on revient dans le traitement de texte avec la feuille mise à jour.

Puissance de l'OLE

La puissance de l'OLE apparaît mieux illustrée par un autre exemple. Supposons que l'on souhaite associer une remarque vocale à un document d'un traitement de texte. On dispose d'un programme pour enregistrer et repasser le son (appelons-le RECORDER), mais le traitement de texte ne connaît rien du format des sons. On peut enregistrer la remarque vocale avec RECORDER et la donner au traitement de texte. Celui-ci conserve les données dans le document sous deux formats : le son numérisé (qu'il ne comprend pas) et un format d'affichage (par exemple, une icône symbolisant une bouche). Le traitement de texte affiche cette icône. Quand elle se trouve sélectionnée, dans le traitement de texte, le programme RECORDER est invoqué (il n'a pas besoin d'être visible), le traitement de texte lui envoie le son numérisé et RECORDER l'exécute sur le haut-parleur.

Ainsi, grâce à l'OLE, les applications peuvent devenir des outils spécialisés. Un simple document est susceptible de contenir diverses données, ayant chacune leur format spécifique, chaque format étant associé à une application particulière. Comme vous devez vous en douter, l'OLE constitue un système complexe reposant sur beaucoup de nouveaux concepts.

Des dialogues organisés

Pour les utilisateurs, l'un des éléments les plus familiers de Windows est la boîte de dialogue Fichier/Ouvrir. Ils savent qu'en tapant ALT-F-O, ils ouvriront toujours une boîte de dialogue présentant une liste des fichiers qui peuvent être chargés dans une application.

Pour les programmeurs, la construction d'une boîte de dialogue Fichier/Ouvrir représente une véritable épreuve de force. Cela exige de nombreuses opérations d'analyse de chaîne et de validation auxquelles on peut échapper avec la plupart des autres boîtes de dialogue. La majorité des programmeurs pensent que la boîte de dialogue Fichier/Ouvrir est si souvent utilisée qu'elle devrait être fournie par Windows lui-même. Eh bien, c'est fait. La nouvelle bibliothèque COMMDLG.DLL exporte 9 fonctions capables d'offrir un gain de temps considérable. On y trouve non seulement une fonction GetOpenFileName, qui invoque une boîte de dialogue Fichier/Ouvrir, mais également GetSaveFileName pour une boîte Fichier/Sauvegarder. Une autre boîte couramment employée est destinée à l'impression, et COMMDLG fournit une fonction PrintDlg.

Pour que les utilisateurs puissent personnaliser les couleurs et les polices dans une application, COMMDLG propose deux autres boîtes de dialogue standard: ChooseColor et ChooseFont. La première permet de sélectionner et de personnaliser les couleurs d'après les modèles RGB (rouge vert bleu) ou HLS (pour Hue Lightness Saturation). La seconde peut distinguer les polices TrueType des autres et, si on le souhaite, limiter la sélection aux polices TrueType. Pour les traitements de texte et pour les éditeurs, COMMDLG offre également les boîtes FindText et ReplaceText.

Au premier abord, on peut trouver ces boîtes de dialogue assez complexes, en raison des nombreuses options de personnalisation disponibles. Mais il est bien plus facile de s'en servir que d'écrire soi-même le code des procédures.

La fonction Escape

La fonction Escape de Windows a toujours semblé un peu bizarre. Bien sûr, il y a des fois où l'on souhaite contourner la GDI pour atteindre directement le pilote d'imprimante afin de profiter des caractéristiques d'une machine qui n'est pas supportée directement par la GDI. Mais Escape est également nécessaire pour l'impression normale — pour commencer et arrêter l'impression et spécifier les limites de page.

L'une des difficultés de la fonction Escape est que les paramètres employés pour passer des données entre une application et un pilote d'imprimante sont déclarés sous la forme de pointeurs de chaîne. Tout le reste doit être converti en pointeur de chaîne long, interdisant la vérification de type en C. Windows

PC TECHNOLOGIE ANNONCE

Six DRIVER

LE PRODUIT QUI VA REVOLUTIONNER LA GESTION DES INDEX!

1 950 F HT
(Offre Spéciale)

- Créer vos index sur mesure
- Déplacement linéaire dans l'index
- Blocs de champs mémo définissable par l'utilisateur

- Index compact
- Index compound
- Index conditionnel
- Construction de sous-index

- Relecture d'enregistrements
- Compatibilité FOXPRO 2
- Moteur EXPERT HELP
- Autre commandes et fonctions

Je désire recevoir la documentation concernant le produit Six DRIVER

Société .....

Nom ..... Fonction .....

Adresse .....

.....

Code Postal ..... Ville .....

Téléphone ..... Télécopie .....

A retourner à : PC TECHNOLOGIE - 4/4 bis, Rue Bourgeoise - 92800 PUTEAUX
Tél. : (1) 42.04.00.88 - Fax : (1) 42.04.69.80

PC TECHNOLOGIE

Service lecteurs n° 24 (dernière page)

3.1 contient 6 fonctions destinées à l'impression sans Escape, plus une nouvelle fonction d'impression. Les nouvelles fonctions StartDoc, SetAbortProc, AbortDoc et EndDoc remplacent les fonctions Escape correspondantes. On peut maintenant utiliser StarPage et EndPage à la place des appels NEWFRAME et BANDINFO. Bien entendu, les fonctions Escape substituées sont toujours supportées pour des raisons de compatibilité avec les versions précédentes.

Citons encore ResetDC parmi les dernières fonctions de la GDI. Elle permet de changer le mode d'imprimante, mais on ne peut pas la solliciter entre les appels à StartPage et à EndPage. Elle sera normalement employée pour changer l'orientation de la page (portrait ou paysage) au cours de l'impression, chose impossible auparavant.

Au fil des ans, Escape est devenue une sorte de catalogue de tout ce que la GDI ne peut pas faire. Heureusement, la GDI améliorée de l'API de Windows 32 bits nous promet d'éliminer la plupart de ces fonctions. Windows 3.1 montre la voie en supprimant la fonction Escape des routines d'impression.

Si vous essayez de former un programmeur parfait, vous voudrez sans doute soumettre cette étrange créature au stress afin de voir comment elle réagit. Pour un programmeur, le stress peut être provoqué par un écran tremblotant, par des compilateurs C qui indiquent des erreurs de mémoire de pile à trois heures du matin, par des vendeurs de pizzas qui oublient d'ajouter le fromage ou par la documentation de l'OLE.

Si vous tentez de développer une application parfaite, vous devez aussi la soumettre au stress — peu (ou pas) de mémoire, pas d'espace disque disponible ou pas de handles de fichier. La nouvelle bibliothèque STRESS.DLL contient des fonctions qui recréent artificiellement ces conditions pour que vous puissiez tester le comportement de votre programme. Destinée au test, elle ne sera fournie qu'avec le SDK de Windows 3.1. STRESS.DLL exporte 10 fonctions.

AllocDiskSpace crée un grand fichier, réduisant ainsi la quantité d'espace libre sur disque à une valeur spécifiée. AllocFileHandles alloue des handles de fichier, n'en laissant également de disponibles qu'un nombre déterminé.

STRESS.DLL contient trois fonctions pour le test des conditions de saturation de mémoire. AllocMem attribue de la mémoire globale, AllocUserMem octroie de la mémoire dans le tas local de USER.EXE (où sont stockées les informations concernant les ressources et les fenêtres). AllocGDIMem accorde de la mémoire dans le tas local de GDI.EXE (où se trouvent stockés les objets GDI). Avec chacune de ces trois fonctions, on spécifie la taille d'un bloc compact de mémoire qui doit être indisponible. Cinq autres fonctions de STRESS.DLL rétablissent une situation normale.

Analyse du stress

J'ai la curieuse impression que le fait d'analyser celui d'un programme augmente en fait le stress du programmeur. Par exemple, quand avez-vous vérifié pour la dernière fois si un appel à CreatePen échouait? Savez-vous seulement ce qui se passe alors? Pour le découvrir, il suffit d'appeler AllocGDIMem avant CreatePen. Vous verrez qu'il ne se produit rien de très grave. CreatePen retourne la valeur NULL, bien sûr, et si vous sélectionnez le handle nul dans le contexte de périphérique avec la fonction SelectObject, celle-ci renvoie de nouveau NULL pour indiquer une erreur. Le contexte conserve l'ancien crayon.

Une autre manière d'employer la bibliothèque STRESS.DLL consiste à écrire un petit programme utilisant un timer Windows pour générer aléatoirement des conditions de saturation de la mémoire ou du disque. On peut lancer ce petit programme à chaque fois que l'on voudra tester une application en développement. Cela devrait amuser vos testeurs bêta.

Qu'est-ce qui se passe réellement à l'intérieur de Windows? Auparavant, les programmeurs qui avaient besoin de connaître certaines données internes de Windows devaient s'en remettre à des méthodes non documentées. La nouvelle bibliothèque TOOLHELP.DLL exporte 38 fonctions permettant au développeur d'examiner tout l'environnement Windows et d'implémenter plus facilement des débogueurs. Grâce à TOOLHELP, on peut obtenir des informations sur toutes les tâches et sur tous les modules exécutés sous Windows, sur toutes les classes de fenêtre couramment enregistrées, sur tous les blocs de mémoire globale et sur les tas locaux associés aux applications et aux DLL. Le support de débogage inclut le changement de tâche, le verrouillage des entrées, la gestion des interruptions et des exceptions.

Après avoir lutté avec les nouvelles API de Windows 3.1, il faut faire une pause. Elle est autorisée par le drag-and-drop (tirer et relâcher, curieusement appelé «glisser-déplacer»). Il s'agit d'une API très simple composée de 4 nouvelles fonctions et d'un nouveau message. L'API drag-and-drop permet à l'utilisateur de saisir un fichier (ou un groupe de fichiers) dans le Gestionnaire de fichiers et de le «tirer» jusqu'à votre application. Les noms des fichiers sont fournis à votre programme.

Ce dernier commence par appeler DragAcceptFiles (en général pendant l'initialisation de la

fenêtre) pour indiquer que la fenêtre peut accepter les fichiers tirés du Gestionnaire. Ensuite, lorsque l'utilisateur tire un fichier du Gestionnaire de fichiers vers la fenêtre de l'application, cette dernière reçoit un message WM_DROPFILE. Le paramètre wParam du message est un handle de «drop» (relâchement) employé par les trois autres fonctions. Le programme utilise ce «handle de drop» pour appeler DragQueryFile — plusieurs fois si nécessaire — afin d'obtenir les noms des fichiers relâchés. La fonction DragQueryPoint obtient les coordonnées du curseur de la souris relativement à la fenêtre de l'application.

Une fois cela terminé, l'application appelle DragFinish pour désallouer la mémoire employée pendant l'opération drag-and-drop. C'est une API si pratique et si simple qu'elle devrait se trouver intégrée à tous les programmes qui se servent des fichiers.

Aide à l'installation

Les programmes Windows deviennent de plus en plus complexes, et donc plus gros. De nos jours, il n'est pas surprenant de voir des logiciels occuper une demi-douzaine de disquettes, ou davantage. Pour réduire cette quantité, plusieurs éditeurs distribuent leurs produits sous forme de fichiers compressés. Au cours de l'installation, ces derniers sont décompressés pour être stockés sur le disque dur.

Le SDK de Windows 3.1 facilite cette opération en offrant les programmes DOS intitulés «COMPRESS.EXE» et «EXPAND.EXE». COMPRESS réduit les fichiers à l'aide de l'algorithme de compression Lempel-Ziv, tandis que le programme EXPAND leur rend leur forme initiale.

Les vendeurs qui souhaitent développer un programme d'installation pour Windows peuvent donc compresser leurs fichiers pour les stocker sur disquettes, puis profiter de la bibliothèque LZEXPAND.DLL pour l'installation sur disque dur. Cette DLL exporte 10 fonctions qui implémentent les algorithmes de décompression Lempel-Ziv.

Par exemple, pour décompresser un seul fichier, le programme d'installation peut ouvrir le fichier source compressé et le fichier destinataire en appelant LZOpenFile. Un appel à LZCopy copie le fichier tout en le décompressant. Le programme d'installation appelle ensuite LZClose pour fermer les deux fichiers. LZEXPAND dispose également de fonctions pour les copies de fichiers multiples ou pour opérer sur des fichiers contenant à la fois des données compressées et décompressées.

Windows 3.1 possède une base d'enregistrement qui conserve des données relatives aux applications installées sous Windows. Son format ressemble beaucoup à celui des informations des fichiers INI, mais avec une hiérarchie multiniveau plus souple. Elle facilite l'intégration des applications via DDE et OLE et aide le Gestionnaire de fichiers et le Gestionnaire d'impression à obtenir des informations sur d'autres programmes. Sept fonctions sont consacrées à l'API d'enregistrement. Il y a encore dans Windows 3.1 quelques améliorations concernant la chaîne de commande DDE utilisée pour installer des applications dans un groupe de programmes du Gestionnaire de programmes.

Lorsque plusieurs programmes partagent une DLL, des difficultés peuvent survenir au moment où celle-ci est mise à jour. Un utilisateur risque d'installer une application qui remplace une version d'une DLL par une version précédente. Pour éviter ces problèmes, une nouvelle API «version» aide à l'identification des versions de DLL.

Passer à la version 3.1?

Windows 3.0 a eu un tel impact sur l'industrie du PC que les développeurs DOS et Windows ont été vivement encouragés à adapter leurs programmes à cette version. Cependant, malgré l'arrivée de Windows 3.1, de nombreux utilisateurs continueront sans doute à travailler avec Windows 3.0.

Cela pose un dilemme aux développeurs. Faut-il aller de l'avant et profiter tout de suite des caractéristiques de Windows 3.1? Ou faut-il attendre?

Heureusement, la réponse n'est pas aussi catégorique. Il est possible de tirer parti de ces caractéristiques et de faire tourner ses programmes sous Windows 3.0. Cela tient à la magie de la liaison dynamique. On peut employer l'OLE, la DDEML, les nouvelles boîtes de dialogue et les vérificateurs de version si les DLL requises accompagnent une application.

L'étape la plus importante consiste à vous assurer que vos programmes Windows 3.0 peuvent utiliser TrueType. Cela ne devrait pas exiger de grosses modifications — et même pas du tout — à moins que vous n'ayez l'intention de remplacer votre boîte de sélection de polices par celle de COMMDLG.DLL. Inutile de prévoir des vacances pour l'instant. Dès que vous aurez assimilé Windows 3.1, il sera temps de vous pencher sur Windows 32 bits! ❐

Cet article est extrait de la revue Microsoft et a été traduit par Henry-Luc Planchat.

LANGAGE

LE C++ SANS OBJET

Ou comment tirer profit du C++ sans connaître la POO

RICHARD HALE SHAW

Le langage C ne peut être étendu qu'en écrivant des fonctions et des définitions de structures nouvelles. C++, pour sa part, permet en outre à l'utilisateur de définir des types de données et des nouveaux opérateurs associés. C'est ainsi que l'opérateur «+» peut avoir un sens différent selon le type de données employé comme opérande.

Par exemple, C++ contient plusieurs objets standards pour «stream input» et «output» «cin», «cout» et «cerr». Il est important de noter que l'emploi du mot «objet» est ici indispensable : c'est la seule manière de décrire «cin», «cout» et «cerr». En outre, l'un des avantages de la programmation orientée objet est de permettre d'utiliser n'importe quel objet dès que l'on sait ce qu'il fait - même sans savoir comment.

Ces trois objets correspondent approximativement au «stdin», «stdout» et «stderr» du C, à la différence près qu'ils possèdent également des possibilités qui leur sont particulières.

Comme indiqué plus haut, le C++ permet à l'utilisateur de définir à la demande de nouveaux types de données avec leurs opérateurs associés. Avec «cin», on peut écrire :

```
int c;
cin >> c;
```

au lieu de :

```
int c;
c = getchar();
```

Employée avec «cin», la fonction de déplacement de bit «>>» en C devient celle de l'opérateur «get from» : il détecte un caractère dans le flux d'entrée (input stream) et le place en «C».

Pour le reste, «cin» agit comme «stdin». L'instruction «cout» est l'opposé de «cin» et s'apparente au «stdout» (standard output stream). L'opérateur «<<» («put-to») peut s'utiliser pour envoyer des données à «cout»,

```
printf («Bonjour à tous»);
```

peut se remplacer par :

```
cout <<«Bonjour à tous »;
```

Ces opérateurs ont alors la même fonction que les opérateurs de redirection«<« et »>» sous MS-DOS, OS/2 et Unix.

Dans la commande DOS suivante

```
SORT<DATA.TXT>NEWDATA.TXT
```

l'opérateur «<» envoie à la commande Sort le contenu du fichier DATA.TXT, et l'opérateur «>» expédie le résultat de Sort dans le fichier NEWDATA.TXT.

De manière analogue, on peut donc dire que «<<» envoie à «cout» et «>>» dirige le résultat de «cin» vers une autre fonction ou variable. «Cin», «cout» et «cerr» ne sont pas seulement des abréviations pratiques et faciles à lire tandis que les fonctions équivalentes «getchar» et «putchar» se servent des «ints» et qu'il faut spécifier aux fonctions «printf» et «scantf» le format qu'elles doivent employer, «cint», «cout» et «cerr» déterminent automatiquement le format à utiliser, en fonction du type de variable présent dans le programme.

C'est ainsi qu'on peut écrire :

```
cout<<«Hello»;
```

mais également :

```
cout<<'H';
cout<<'e';
cout<<'l';
cout<<'l'; cout<<'o';
```

A noter toutefois qu'il n'est pas possible d'utiliser les codes ASCII correspondant aux lettres, ceux-ci étant en effet automatiquement identifiés comme des nombres. Il faut se servir de la mention «(char)».

Alors que :

```
cout<<72;
cout<<100;
cout<<107;
cout<<107;
cout<<110;
```

donne 72100107107110.

En revanche;

```
cout<<(char)72;
cout<<(char)100;
cout<<(char)107;
cout<<(char)107;
cout<<(char)110;
cout<<'\n';
```

donne bien «HELLO».

Nul besoin de comprendre la programmation orientée objet (POO) pour tirer parti des avantages du C++. L'un de ses principaux atouts consiste d'abord à être un «meilleur C». Cet article s'intéresse aux possibilités du C++ dont l'utilisateur pourra tirer profit sans faire appel à des classes et sans rien savoir de la POO. A la longue cet utilisateur pourra maîtriser celle-ci, tout en exploitant le C++ grâce à ses connaissances en C.

La combinaison de «cin», «cout», «<<» et «>>» remplace «printf» et permet de faire plus. La routine classique demandant son nom à l'utilisateur pour l'afficher après «Bonjour» s'écrit de la manière suivante :

```
#include<iostream.h>
void main(void){
char nom[30]
cout<<<Votre nom?»;
cin>>nom;
cout<<<Bonjour, »<<nom;}
```

La dernière ligne emploie l'opérateur «put-to» pour concaténer «Bonjour» et la variable «nom» dans «cout». Cet opérateur permet donc de concaténer des éléments de types différents.

Chaque fois que «<<» est utilisé avec «cout», l'affichage continue depuis la dernière position du curseur. Par exemple :

```
#include<iostream.h>
int line;
char filename[80], letter;
printf(«La ligne
%d du fichier
%s contient la lettre '%c'\n»,
line, filename, letter);
```

se remplace par

```
#include<iostream.h>
int line;
char filename[80], letter;
cout<<<La ligne »<<line<<»du
fichier »<<filename<<»contient
la lettre'»<<letter<<»'\n»;
```

Il n'est donc plus nécessaire de formater strictement les données avec '<<', elles s'écrivent telles qu'elles sont entrées. «Cin»,«cout», «cerr» et leurs opérateurs associés peuvent être utilisés dans n'importe quel programme C++.

Il suffit d'inclure l'en-tête IOSTREAM.H. Il est possible de les employer avec des redirections d'entrée-sortie comme on le ferait avec «printf». En tant que tels, ils ne font pas plus partie de C++ que l'utilisation élargie de «>>» et de «<<». Ces objets sont exécutés comme des classes C++, et les fonctions des opérateurs se trouvent remplacées par les nouvelles définitions lorsqu'ils sont utilisés avec des objets de ces classes.

Les commentaires en C++

Au format standard C (/*...*/), toujours présent, C++ ajoute le format de type «commentaire d'une seule ligne» semblable à celui employé en assembleur, à l'exception près que le point-virgule est remplacé par «//».

Exemple :

```
void main(void){
/*ceci est un commentaire*/
cout<<<Bonjour à tous\n»;
// et ceci un autre commentaire
}
```

Avec ce format de commentaire, tout ce qui suit les «//» jusqu'à la fin de la ligne est considéré comme commentaire. On ne peut donc s'en servir pour insérer un commentaire dans une expression. Si le commentaire fait plusieurs lignes, chacune doit commencer par «//». Ce sont les prototypes du C++ qui ont donné l'idée au comité Ansi de les incorporer dans le C.

Les prototypes de fonction obligent le compilateur à effectuer un contrôle plus strict des types en lui retournant le type de donnée résultant d'une fonction, ainsi que le nombre et le type de ses arguments d'entrée. Alors que le compilateur C autorise l'omission des prototypes (tout en émettant un avertissement), C++ impose que chaque fonction ait le sien.

Le programme, HELLO.C

```
void main(void){printf(«Bonjour
à tous!»);}
```

est compilé sans erreurs en Borland C++. S'il est renommé en HELLO.CPP, Borland C++ envoit un message d'erreur indiquant que «printf» n'a pas de prototype de fonction. C'est un inconvénient, mais il apparaît compensé par la certitude que les types des arguments sont vérifiés.

Le prototype de fonction du C++ présente la particularité, par rapport à celui du C, de pouvoir fournir des arguments par défaut. Autrement dit, un prototype de fonction peut inclure des valeurs par défaut pour les paramètres d'une fonction.

Prenons, par exemple, le listing de la fonction WINDOW.CCP. Les coordonnées et le pointeur de message des quatre fenêtres sont donnés par défaut dans le prototype de fonction.

En revanche, alors que le premier appel de fenêtre messagewindow(); utilise les coordonnées par défaut, le second appel les précise :

```
messagewindow(150,200);
```

A noter cependant qu'il est impossible de laisser un argument par défaut et de préciser les autres.

Par exemple :

```
messagewindow(,200,200,560);
//FAUX !
```

où le premier argument pris par défaut ne sera pas compilé. A souligner que les arguments choisis par défaut constituent un outil précieux dans la définition des objets. Ils permettent de créer des fonctions utilisant en général le même jeu d'arguments, tout en autorisant des exceptions.

Et les fonctions qui sont passées d'un objet à l'autre peuvent le plus souvent employer des arguments par défaut. Cependant, lorsque la fonction est appelée d'un objet dérivé, celui-ci peut lui passer des arguments différents si nécessaire.

Filtrage de fonctions

Le C++ permet de créer plusieurs versions de la même fonction à l'intérieur d'un même programme. Si cette possibilité semble présenter un intérêt restreint au premier abord, elle permet, par exemple, de simplifier l'utilisation des fonctions d'heure. La librairie du C en comporte toute une série, les unes employant «struct tm», d'autres «time_t» (toutes deux définies dans TIME.H). Le passage de l'une à l'autre peut compliquer sensiblement l'écriture d'une application. On peut contourner cette difficulté en créant une seule fonction d'heure qui renvoie le pointeur approprié.

Par exemple, le prototype de la fonction d'heure :

```
time-t(time_t *timer);
```

peut coexister avec

```
struct tm*time(void){
time_t t=time(NULL);
return localtime(&t);}
```

(voir le listing de TIME.CPP).

L'éditeur de liens se servira de la fonction la plus adéquate en C++, le type des arguments fait partie du nom de la fonction, au même titre que son identifiant. Il est impossible de faire cela en C, où un nouveau nom de fonction remplace l'ancien dans son espace d'application.

Deux fonctions ayant le même identifiant ne peuvent coexister dans le même espace d'application. Le listing de STR.CCP indique comment employer le même nom de fonction, «strcpy», pour copier une chaîne de caractères soit avec la liste d'arguments de «strcpy», soit avec celle de «strncpy».

L'addition de la longueur de l'argument détermine quelle version utiliser. A noter que les fonctions filtrées sont également utiles pour la définition d'objets; elles peuvent servir pour permettre à un objet de se servir de plusieurs types de données. Il est même possible de filtrer le constructeur de classe, donnant ainsi la possibilité de créer un objet utilisable dans plusieurs ensembles de circonstances.

Turbo C permet l'utilisation d'«intrinsics» —le compilateur insérant alors le code de la fonction dans celui de cette dernière. Ceci peut allonger le code de l'application, mais la suppression du temps pris par la fonction d'appel rend l'exécution plus rapide. De surcroît, si le code de la fonction est plus court que celui qui appelle celle-ci, le code de l'application sera même plus court. C++ fait encore mieux avec le mot-clé «inline». Utilisé dans une définition de fonction, il spécifie que, lorsque cela est possible, le compilateur doit placer une copie du code de la fonction dans celui de l'application chaque fois que la fonction est appelée.

Par exemple :

```
inline void error-exit(int
errval, char *message){
printf(«Erreur        %d\n»,
errval,message);
exit(errval);
}
```

définit la fonction «inline error_exit». Son code est inséré dans celui de l'application à la place de l'appel de cette fonction. En étudiant le listing de INLINE.CPP puis celui d'INLINE.ASM, code de son assemblage sous Borland C++, on s'aperçoit que si le résultat de «fopen» demande un appel d«'error_exit», le code de cette fonction est inséré dans le code principal.

Par rapport aux macros du C, les fonctions «inline» présentent l'avantage que le compilateur C++ vérifie les arguments passés et retournés de manière aussi stricte que pour les autres fonctions. Le compilateur C++ n'a pas à implanter les fonctions «inline» dans le programme.

A relever que la définition d'une fonction «inline» doit précéder tout appel qui lui est fait. Si la fonction doit être utilisée de façon intensive, il est préférable de placer sa définition dans le fichier d'en-tête.

Pour terminer, il y a quelques précautions à prendre lorsque l'on utilise des fonctions C (traitées par un compilateur C) dans un programme C++. C vérifie des types dans les fichiers source qui partagent le même prototype de fonction. Mais cette vérification n'est pas assurée entre des modules objets compilés séparément.

Il faut un prototype dans chacun des fichiers source qui appellent la fonction. C++ emploie, en revanche, une édition de liens de type «type-safe linkage», il contrôle la confor-

mité des types si une fonction est appelée dans un module différent de celui où son prototype a été défini et ajoute un suffixe au nom de la fonction qui identifie le type des paramètres.

Ce procédé, appelé «mangling» (que nous traduisons par mixage), permet à l'éditeur de liens de traiter correctement les appels à une même fonction dans différents modules. Les compilateurs C ne mixent pas les noms des fonctions, il n'est donc pas possible de les utiliser tels quels dans un programme C++. On ne peut pas non plus les filtrer. Comment donc indiquer au compilateur C++ qu'une fonction doit être compilée avec les conventions d'édition de liens du C? A l'aide de la spécification suivante :

```
extern «C»{ #include<stdio.h> }
```

La déclaration «extern »C«» permet au C et au C++ de coexister, les fonctions traitées par le compilateur C étant accessibles au C++ sans avoir besoin d'être retraitées. La plupart des compilateurs C++ sont en fait des préprocesseurs; avec un compilateur C le préprocesseur traduit le code C++ en code Ansi C, le code résultant étant alors «préprocessé» et traité par le compilateur C. Le C++ de Borland possède en revanche un vrai compilateur, un seul macroprocesseur prépare le code, et les codes C et C++ se trouvent compilés. Les règles du C sont utilisées pour les fichiers ayant une extension C et celles du C++ pour les CPP.

Il est également possible de se servir de l'option de ligne de commande -P pour s'assurer que seules les règles du C++ sont utilisées au cours de la compilation. La déclaration «extern »C«'» évite alors le mixage des fonctions qui y sont spécifiées.

Habituellement, la spécification de l'édition de liens est placée dans le fichier d'en-tête, avec les prototypes des fonctions C. Le fichier d'en-tête STDIO.H du C++ de Borland contient donc la déclaration «extern »C«» sous la forme suivante :

```
#ifdef_cplusplus
extern «C»{
#endif ... #ifdef_cplusplus
}
#endif
```

«_cplusplus» est défini lors de la compilation du programme C++. Ceci ne concerne pas les fonctions standards du Turbo C ou du C++ de Borland, seulement celles qui ont des prototypes dans le fichier d'en-tête. ❐

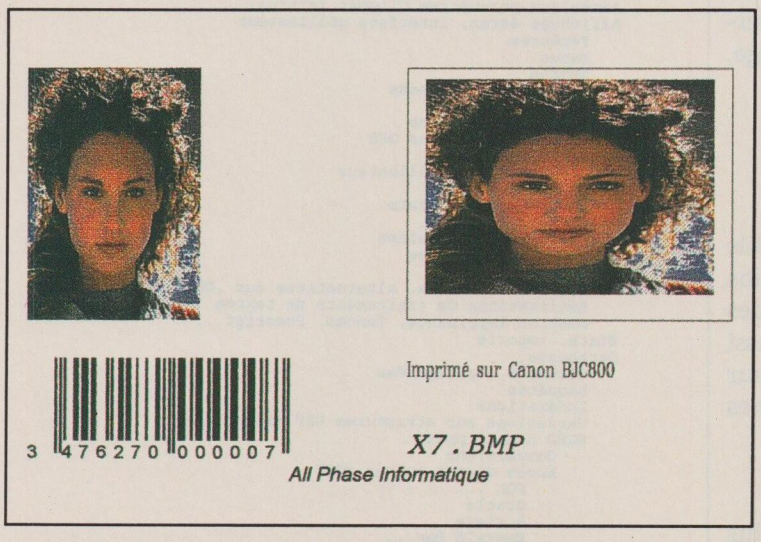


Image capturée avec PICBMP, textes, images et graphiques entièrement réalisés avec ces DLLs.

DLLs, des fonctionnalités puissantes pour Visual Basic, dBFast, ObjectVision, Excel....

PICBMP (PAL INPUT CAPTURE BMP)

Vous possédez un caméscope, Ion, magnétoscope, caméra... Quelques clics sur la souris, l'image statique ou en mouvement est sous forme de fichier BMP sur votre disque. L'ensemble comprend :
Une carte vidéo (driver Windows fourni) et de numérisation 32768 couleurs, Temps réel, 100% TARGA 16 compatible, accepte tout signal PAL RGB ou composite (deux entrées)
Un logiciel puissant et simple (100% Windows), effets en temps réel de zoom, contrôle du signal vidéo (contraste, offset, couleurs...). Retailler l'image ...

DLL Images BMP (LDIBSHOW.DLL)

Ajoutez à votre langage favori de nouvelles fonctions qui vont vous permettre de manipuler des images BMP du simple affichage aux animations les plus complexes (zoom, déformations, effets spéciaux...)

Primitives Graphiques (LGRPRINT.DLL)

Réalisez n'importe quel tracé sur écran, imprimante, imageur, table traçante...en passant vos données à la DLL! Dessins scientifiques, statistiques, organigrammes, formulaires, badges, codebars.
Véritable langage de description de page vous donnant accès aux fonctions vectorielles et Bitmap.
Vous gérez 16 millions de couleurs, précision au 10ème de mm sur 3,20m x 3.20m..

All Phase Informatique

34 rue Jean Pierre Bredy 69100 Villeurbanne
Tél 78 89 20 60 Fax 78 89 21 26

Visual Basic, Excel et Windows sont des marques déposées de Microsoft, dBFast de Computer Associates et ObjectVision de Borland, Targa est une marque déposée.

Service lecteurs n° 25 (dernière page)

LANGAGE

NANTUCKET TOOLS II
Augmenter la puissance de Clipper

GILBERT CHAUVAUX

L'utilisateur Clipper dispose d'un vaste éventail d'outils. En a-t-il besoin? A priori, son seul investissement de départ lui permet, dans la plupart des cas, de réaliser ses applications. Au besoin, il peut accomplir certaines routines lui-même en langage Clipper ou avec les langages de plus bas niveau C ou assembleur.

Il dispose avec Clipper d'interfaces documentées de Nantucket (le système Extend que nous étudierons par la suite). Les outils sont nécessaires pour des fonctions complexes : utilisation du mode graphique, optimisations, exploitation des particularités «internes» de Clipper, etc. Cette exploitation permet, par exemple, d'étendre le système de programmation objet de la version 5.0 et de créer de nouvelles classes avec héritage, aussi performantes que les classes natives.

Embarras du choix

L'utilisateur Clipper peut se procurer quantité d'outils généralement à bon marché et performants. Les gains potentiels sont importants, ils améliorent la qualité des applications en termes d'interface utilisateur, mais aussi la rapidité d'exécution, la productivité, etc. L'utilisateur devra en contrepartie garder à l'esprit les problèmes suivants :

- choix de produits parfois complexes (librairies),
- coût d'apprentissage du produit,
- risques de dépendance au fournisseur qui sera le plus souvent une petite société avec des concepteurs très «remuants»,
- risque de dépendance à la version de Clipper (la version 5.0 a invalidé la plupart des librairies au moment de sa sortie).

Quels outils ?

Il existe aux Etats-Unis, en Angleterre, en Allemagne et en France une multitude de produits dédiés à Clipper ou en rapport avec celui-ci. Il serait trop long d'en faire la liste.

```
Sujets

   Aides à la réalisation, optimisation...
      générateurs
      Environnements, outils Case, dictionnaires...
      Optimisation
         Linkeurs
         Gestion mémoire
      gestion versions, envoi de versions...
      Programmation objet : outils de base, création de classes
      Décompilateurs -reverse engineering, documenteurs

   Fonctions utilisées dans les applications
      Accès aux fonctions DOS, Bios ...
      Accès aux paramètres Clipper (status)
      Affichage écran, interface utilisateur
         Fenêtres
         Menus
         Souris
            Fonctions de base
            Boutons...
         Effets esthétiques
         Browsers, systèmes GET
         Graphisme
            Interfaces utilisateur
            Images
            Graphiques, stats
      Gestion de textes
         Traitement de chaînes
         Recherches rapides
         Spell-checking
         Gestion des mémos, alternatives aux .DBT
         Réalisations de traitements de textes
         Gestion imprimante, fontes, Poscript...
      Etats, reports
      Databases
         Sécurité, sauvegardes
         Requêtes
         Indexations
         Variations sur structures DBF (Flexfile)
         SGBD non-Clipper
            Conversions
            Accès direct autres SGBD
               SQL ...
               Oracle
               Btrieve
               Emerald Bay...
      Mathématiques, statistiques, calculs math. et calendaires
      Outils bureautiques: calculateurs, agendas
      Fonctions réseaux locaux
      Communications
      Périphériques non standards
         Lecteurs de codes à barres
         Scanners
    etc.
   -
```

Il est, en revanche, possible d'effectuer une classification par sujet :

Il existe une autre classification, selon le mode de réalisation de la fonction :

Une des grandes forces du langage Clipper 5 réside dans son architecture ouverte. Son concepteur, Nantucket, l'a conçu pour recevoir des extensions sous forme de routines en C, en assembleur ou par le biais de librairies. Il propose ainsi Nantucket Tools II qui comprend près de 550 fonctions qui élargissent considérablement le champ d'application de Clipper.

```
- fonctions appelées en Clipper, simples et indépendantes
- fonctions complexes, systèmes de fonctions couplées dépendantes
- Arborescence de classes
- système de programmation complet : POO, méta-langage
- applications complexes fournies en sources Clipper à modifier
  et à insérer dans votre application,
- pilote à chaîner avec l'application ou résidant à l'exécution
- programme indépendant (stand-alone) : linkeur, générateur,
      convertisseur,...
```

Nantucket Tools II

La Nantucket Tools version II a été élaborée par la filiale allemande de Nantucket. Cette importante librairie est généraliste, mais avec des débordements dans des spécialités (réseau, communication). Elle bénéficie de l'apport technique et du support de Nantucket, ce qui n'est pas négligeable.

Les fonctions seront présentées dans l'optique d'une programmation «style 5.0» utilisant toute les particularités de Clipper 5 (la librairie comprend beaucoup de fonctions utiles en Summer87 qui sont maintenues pour assurer la compatibilité avec Nantucket Tools I). Nantucket Tools II comporte 546 fonctions allant de la simple fonction de modification de chaîne à la compression et à l'envoi de fichiers avec protocoles. Ces fonctions sont réparties en 17 chapitres, que nous détaillerons par la suite.

Elles sont livrées dans une librairie de 218 Ko et un .OBJ de 28 Ko (pour les pilotes de périphériques). 15 fonctions ayant trait au système GET et une fonction de fenêtre sont fournies en sources Clipper (3 ne sont pas documentées), dans 8 fichiers .PRG.

Le package comprend également

- 11 fichiers headers .CH associés aux fonctions,
- 26 programmes exemples .PRG,
- 4 utilitaires fournis en sources Clipper, avec procédures Rmake (.RMK) :
  - utilisation des palettes EGA/VGA
  - modification et création des fontes de caractères écran (avec documentation et 31 fichiers fontes).
  - des benchmarks de mesure de vitesse du micro,
  - un programme complet de formatage de disquettes,
- une procédure d'édition de liens RTLINK pour créer une librairie «prélinkée» avec les pilotes de périphérique,
- un fichier d'aide Norton de 858 Ko,
- un système d'aide complémentaire pour rechercher des mots-clés de 1 450 Ko! L'installation complète (PLL générée) prend environ 3,5 Mo sur le disque dur, dont seulement 300 Ko sont indispensables (.LIB , .OBJ et .CH).

La documentation

Nantucket est documenté par les Norton et par le système complémentaire, ainsi que par trois manuels principaux. Que ce soit dans ces manuels ou dans les Norton, la difficulté de départ est de choisir le chapitre concerné, ce qui implique de déterminer d'abord le «livre», car il n'y a pas de liste unique des fonctions.

```
+----- NANTUCKET TOOLS II for Clipper 5.01 » 1stBook » String Manipulation ----+
|   Expand    Search...    Options    1stBook    2ndBook    3rdBook    Intro   |
|-------------------------------------------------------------------------------|
|CHARREPL()    Replace certain chara|  Window Functions      |                  |
|CHARSORT()    Sort sequences within|  Extended Drivers      |                  |
|CHARSPREAD()  Expand a string at to|  Serial Communication  |                  |
|CHARSWAP()    Exchange all adjoinin|_ String Manipulation   |                  |
|CHARUNPACK()  Decompress (unpack) a+-----------------------+                  |
|CHARXOR()     Join corr. ASCII codes of paired strings using EXCLUSIVE OR      |
|CHECKSUM()    Special checksum algorithm for string.                           |
|COUNTLEFT()   Count a particular character at the beginning of a string.       |
|COUNTRIGHT()  Count a particular character at the end of a string.             |
|CRYPT()       Encrypt and decrypt string.                                      |
|CSETATMUPA()  Determines the setting of multi-pass mode for ATxxxx() functions_|
|CSETREF()     Determines if reference sensitive functions will return a value._|
|EXPAND()      Expand a string by inserting characters.                         |
|JUSTLEFT()    Move characters from the beginning to the end of a string.       |
|JUSTRIGHT()   Move characters from the end of a string to the beginning.       |
|LIKE()        String comparison using wildcard characters.                     |
|LTOC()        Convert a logical value into a character.                        |
|MAXLINE()     Finds the longest line within a string                           |
|NUMAT()       Counts the number of occurrences of a sequence within a string.  |
|NUMLINE()     Number of lines required for string output.                      |
|NUMTOKEN()    Number of tokens in a string.                                    |
+-------------------------------------------------------------------------------+
```

Exemple d'écran Norton faisant apparaître les 3 livres (books) et les chapitres.

NANTUCKET TOOLS II

Après une très brève introduction, les manuels détaillent les fonctions par ordre alphabétique. Pour se faire une idée de leur utilisation, il est nécessaire d'avoir recours aux exemples et aux essais.

Les descriptifs des fonctions sont de bonne qualité. Comme c'est souvent le cas, il y a quelques oublis qui obligent à faire des essais. Il est curieux de constater que les concepteurs de logiciels laissent passer des fautes de documentation qui apparaissent à la première lecture, même sans avoir à employer la fonction. Il faut donc prévoir de passer du temps à comprendre quelle fonction ne sera pas exploitée et laquelle se révélera utile.

Les performances

Les fonctions sont écrites en C et assembleur et s'avèrent performantes. Le seul problème qui peut se poser est l'utilisation de la mémoire. L'analyse du contenu de la librairie fait apparaître que les fonctions sont réparties sur un grand nombre de petits fichiers objets (dans la librairie), souvent à raison d'une fonction par objet. Ceci explique que cette librairie de 218 Ko contient moins de 70 Ko de code et de données présents en mémoire à l'exécution, pour la plupart en zone fixe (sous RTLINK). Ainsi, un programme utilisant les 546 fonctions n'utilise que 65 Ko de mémoire fixe supplémentaire, avec une taille d'exécutable augmentée de 80 Ko.

Les fonctions en question

☛ Les fenêtres (22 fonctions)

Les fenêtres permettent au programme d'utiliser plusieurs «flux» (affichages et saisie) à tour de rôle dans des fenêtres distinctes, les fonctions se chargeant de sauver et de mémoriser les contenus, couleurs, curseurs, etc. Elles peuvent se charger d'une interaction avec l'utilisateur qui souhaite déplacer en bloc les fenêtres (sans modification de dimension) de façon transparente.

Les paramètres de création de la fenêtre permettent d'interdire ou de limiter ces déplacements.

S'il est possible, sinon facile, de réaliser des fenêtres, il est difficile d'atteindre l'efficacité de Nantucket Tools.

Les fenêtres sont en grande partie gérées à un très bas niveau, celui du Bios par redirection de l'interruption 10H. Les programmes lancés par RUN peuvent bénéficier des fenêtres, y compris le RUN COMMAND qui permet de faire du DOS dans une fenêtre, comme sous Windows.

On regrettera l'absence de possibilité de distinguer le contenu de la fenêtre de son apparence et, corrélativement, l'impossibilité de modifier ses dimensions.

Ces fenêtres sont très simples à utiliser pour des affichages annexes déroulement d'une recherche, indexation... Les instructions suivantes suffisent, la fenêtre annexe étant créée :

```
WSELECT(<handle de la fenêtre annexe>)
...appel fonction...
WSELECT(0)   // écran entier (fenêtre implicite)
```

☛ La gestion étendue des périphériques — clavier-écran-imprimante — (51 fonctions)

Ces fonctions concernent principalement :

• la gestion du mode vidéo (texte), par exemple VGA 50 lignes, etc.,

• la demande de gestion clavier par le DOS (redirection possible),

• une possibilité de pause automatique dans les listes :

```
SETLINES(20) LIST
     // pause toutes les 20 lignes
```

• la modification du nombre de lignes/colonnes jusqu'à 255, ce qui permet de prendre en compte des écrans non standards (pleine page, etc.), mais aussi d'avoir un écran virtuel supérieur à 25 x 80. Le défilement est automatiquement géré en fonction de la position du curseur, y compris dans les masques de saisie. On peut donc réaliser des masques à plusieurs pages, simplement en changeant les limites de l'écran.

• la gestion des pages écran (mémoire vidéo),

• les conversions de codes pour le clavier et l'imprimante, autorisant une redéfinition complète,

• la gestion spéciale du clavier améliorant les commandes SET KEY et KEYBOARD :

  - possibilité de «trap» (appel de procédure) : pour toute frappe de touche par simple enfoncement de <ALT> <CTRL> <SHIFT> dès que Clipper s'apprête à lire une entrée clavier,
  - simulation du clavier par «scan code» avec possibilité de simuler n'importe quelle touche — y compris les touches fonctions,
  - extension de LASTKEY() pour lire le clavier «en arrière»...

☛ La gestion des communications (40 fonctions)

Ces fonctions gèrent les communications série au niveau matériel, par interruption. Elles autorisent donc un débit élevé et le parallélisme avec d'autres traitements, en particulier la gestion des données transmises. Les fonctions se chargent du tamponnement des données, des protocoles «logiciels» (XON/XOFF) et «hardware» (RTS/CTS) et facilitent l'envoi de fichiers : fonctions de calcul de CRC, de fabrication de paquets

XMODEM (un programme d'exemple utilisant le protocole XMODEM est fourni). Ce type de protocole est obligatoire pour un envoi de fichiers binaires par modems et par ligne avec régulation de flux.

Les unités COM1 à COM4 sont gérées avec :

- débit de 300 à 19 200 bauds,
- parité, longueur et stops configurables,
- paramétrage des adresses de ports et des IRQ,
- accès aux registres des circuits 8250 (LSR, MSR, MCR, IIR),
- gestion (positionnement, lecture) des signaux modems DTR, DSR, RTS, CTS, RI par accès aux registres ou directement,
- mode background implicite pour les réceptions et optionnel pour les envois,
- gestion de l'insertion de bits 0 pour la transparence CCITT (option),
- au choix du programmeur, injection de touches au clavier dans certaines situations :
  - caractère reçu,
  - buffer de réception plein à 75%,
  - buffer d'émission vide (mode background),
  - interruption par XOFF ou inversion CTS,
  - erreur de transmission.

Le programmeur peut se contenter :

- d'ouvrir l'unité de communication choisie,
- en réception, se faire «prévenir» d'une réception ou d'une situation d'exception par des SET KEY...
- ou tester périodiquement l'état du buffer de réception,
- demander à lire et à traiter les données reçues,
- en émission, tester et alimenter périodiquement le buffer d'émission,
- ou se faire prévenir par SET KEY que le buffer est vide.

Ces modes d'utilisation permettent :

- de traiter les données reçues en parallèle à leur réception, donc sans perte,
- de faire autre chose pendant une émission ou une réception,
- de continuer l'émission dès que le buffer est vide, donc à vitesse maximale,
- de gérer les erreurs.

Ces fonctions sont étayées par des exemples.

Voici un exemple simple de transfert de fichier par connexion directe :

```
* envoi FIC.TXT

COM_OPEN(1, FILESIZE("FIC.TXT"))
// COM1+ taille buffer

COM_INIT(1, 9600, "N", 8, 1)
// configuration

COM_HARD(1,.T.)
// protocole RTS/CTS

COM_SEND(memoread("fic.txt"))
// envoi

COM_CLOSE(1)
// fermeture

*réception
COM_OPEN(1, 64000)
COM_INIT(1, 9600, "E", 8, 1)
COM_HARD(1,.T.)
WAIT "frapper une touche quand transmission
terminée"

memowrit("FIC.TXT",COM_READ(1))
// lecture buffer+ écriture

COM_CLOSE(1)
```

Voici un exemple simple d'émulation de terminal :

```
COM_OPEN(1, 1000)
COM_INIT(1, 9600, "N", 8, 1)

while (nKey:=inkey())2<>7
// arrêt par <esc>


   if nKey<>0    // test clavier

      COM_SEND(nPort, nKey)
      // envoi
   endif

   ?? COM_READ(nPort)
   // lecture buffer (0 à 1000 caractères)

end
COM_CLOSE(nPort)
```

La manipulations de chaînes (88 fonctions)

Le but de ces fonctions est d'optimiser des traitements de chaînes pouvant être faites en Clipper. Il y a, de plus, des fonctions complexes

```
CHECKSUM(<chaîne>)
// somme de contrôle (entier 32 bits)

CHARPACK(<chaîne>)
// compression

CHARUNPACK(<chaîne>)
```

```
// décompression

STRDIFF(<c1>,<c2)
// distance de Levenshtein de 2 chaînes
```

Contrairement à ce que semble indiquer la documentation, les chaînes passées par référence ne sont pas directement modifiées sans recopies. En effet, Clipper 5 le déconseille car d'autres chaînes peuvent être affectées.

Pour être très efficace dans le traitement de «grosses» chaînes, il faut tricher — ce que ne permettent pas les librairies. Nous reviendrons sur le sujet dans le traitement des extensions C et assembleur.

Il y a de nombreuses fonctions de traitement de texte (traitement tabulation, recherche-remplacement...) ainsi que d'analyse syntaxique (parsing), pouvant entraîner des développements intéressants.

☛ Les manipulations de bits et de nombres (28 fonctions)

Ces fonctions permettent, en particulier, le calcul booléen sur des nombres de 16 bits, exprimés comme numériques Clipper ou chaînes hexadécimales. L'instruction suivante teste (par masque AND) le contenu du 5e bit du registre hardware LSR de COM1 :

```
if NUMAND(COM_LSR(1),"20") >0...
```

Deux fonctions intéressantes, quoique faciles à réaliser soi-même en C ou en assembleur, permettent de stocker en fichier les nombres flottants (dans les applications techniques) sans perte de précision, en format interne (format «double» standard du C sur 8 octets, utilisé par Clipper) :

```
FTOC(<nombre>)    // chaîne de 8 car.
CTOF(<chaîne>)    // nombre
```

☛ La Vidéo — mode texte — (60 fonctions)

Nantucket Tools ne traite pas le mode graphique, ni la souris. Néanmoins, on peut jouer sur les polices de caractères écran — même faire des rotations et écrire verticalement.

Dans la mesure où l'on peut redéfinir complètement les fontes par points, on peut tracer une courbe! Un programme démo de tracé est fourni.

Ces fonctions permettent aussi :

- de tester le type de carte vidéo,
- de réinitialiser le mode vidéo après un RUN...
- d'exploiter les palettes de couleurs EGA/VGA : effets de couleurs , fading...
- d'utiliser un second écran Mono et de commuter vers l'un ou vers l'autre,
- de rediriger les affichages vers une chaîne de caractères,
- de traiter directement la mémoire vidéo et ainsi :
  - modifier les attributs (couleurs) de zones sans toucher au texte,
  - modifier les textes sans en changer l'attribut,
  - des recherches de textes sur l'écran , etc.
- d'obtenir de beaux effets d'affichage («descentes» de textes...).

Voici un programme qui affiche en inversion toutes les occurrences du mot Nantucket dans un texte sous Memoedit

```
WOPEN(2,2,16,40,.T.)

WBOX(0)
// MEMOEDIT dans la fenêtre

memoedit(memoread("article.txt"),0,0,13,37,.T.,
"user",80)

function user(mode)
    if mode==0
       SCREENMARK("Nantucket","n/w",.F.,.T.)
       // MARQUAGE en inversion
       KEYSEND(I2BIN(SCANKEY()))
       // attente clavier SANS lecture
    endif
return 0
```

On notera la combinaison SCANKEY/KEYSEND qui permet une attente clavier sans lecture et sans perturber la suite (les SET KEY).

La version actuelle de la librairie a un léger problème avec les attributs et change effectivement ces derniers mais au mauvais endroit...

☛ La gestion de fichiers (50 fonctions)

Il s'agit des fichiers DOS, disquettes, répertoires, etc. classiques.

Voici les plus utiles :

```
DISKCHANGE(<drive>)
     // changement unité: A: B:...
DISKNAME()
// unité courante  : A: B:...

DISKTOTAL(<drive>)
// capacité totale

DISKREADY[W](<drive>)
// unité prête? [en écriture]

DISKSTAT(<drive>)
```

```
// code «status» (type erreur)

VOLSERIAL(<drive>)
// no série disquette DOS 4 et 5

DISKSPEED() DISKTYPE()...
//vitesse et type de disque

DIRMAKE(<dir>)
DIRREMOVE(<dir>)
DIRCHANGE(<dir>) // sans commentaire

DISKCHECK(<drive>) FILECHECK(<fic>)
// calcul checksum
```

Il y a également des fonctions de modification d'attributs, date et heure de fichiers, et de génération de noms de fichiers temporaires (pour les applications réseaux).

Le mode de partage des fichiers du DOS (version 3++) en environnement réseau peut être mis en œuvre par :

```
SETSHARE()
GETSHARE().
```

Il existe des fonctions de copie de fichiers évoluées :

```
FILEAPPEND(<fic1>,<fic2>)
// ajoute fic1 à fic2
```

Plus fort, la copie multi-disquettes de gros fichiers :

```
wait "monter disquette"; n:=1

FILECOPY("BIG.DBF", "A:\BIG.001", .T.)
// Backup mode

DO WHILE FILECOPEN()
// test copie terminée

wait "monter disquette"

FILECCONT("A:\BIG."+NTOC(n++,,3,"0")
// suite ...

ENDDO
RETURN
```

FIND NEXT du DOS pour traiter les noms génériques est un programme merveilleux. Il efface un répertoire (l'argument) et ses sous-répertoires :

```
function XDEL(cPath)
local cfic,cseekenv
// standardisation du path (ajout C:\ ..)

FILEDELETE((cPath:=TRUENAME(cPath))+"\*.*")
// effacement générique
cFic := FILESEEK(cPath+"\*.*", 16,.T.)
// recherche fichiers attrib. DIR

DO WHILE .NOT. EMPTY(cFic)

      IF cFic <> "."
      // ".." et "." sautés
        cSeekEnv := SAVEFSEEK()
        // sauver environnement FILESEEK
        XDEL(cPath+"\"+cFic)
        // appel récursif
        RESTFSEEK(cSeekEnv)
        // rappeler environnement FILESEEK

      ENDIF

      cFic:= FILESEEK()
      // sans argument: FIND NEXT

ENDDO

DIRREMOVE(cPath)
// suppression répertoire
```

return nil

On pourrait réaliser des outils «système» de gestion de disque, avec toutefois le handicap (par rapport au C) de la taille du programme. Cependant, l'idée reste valable :

- pour des applications utilisant un .DBF pourquoi ne pas créer un système de documentation du disque?
- pour prototyper une application qui sera transcrite en C ou en C++.

☛ La gestion de l'impression (15 fonctions)

Ces fonctions permettent :

- de connaître le nombre d'imprimantes,
- de tester leur état (sans se restreindre à LPT1),
- d'imprimer un fichier,
- d'imprimer une hard-copy d'écran avec la possibilité de remplacer les caractères semi-graphiques,
- un test de début de page qui fait gagner une fortune en papier-listing :

```
If ! TOF();  Eject; end
```

- de gérer le spool du DOS (si la commande PRINT a été activée), sans lancer de RUN PRINT.

```
IF SPOOLACTIVE()
```

```
SPOOLADD("C:\TEST\PRINT.DAT")
ENDIF
```

☛ La gestion de dates et d'heures (27 fonctions)

A part les fonctions d'initialisation date et heure, SETDATE() et SETTIME(), ce sont les fonctions classiques de détermination du mois, etc.

A partir d'une date, voici la date d'échéance «15 jours fin de mois» :

```
ADDMONTH(BOM(date()),1)+14
```

☛ Les bases de données (7 fonctions)

Ces fonctions qui fournissent des informations sur les champs n'ont pas d'intérêt en Clipper 5. En revanche, deux fonctions donnant la taille «logique» et actuelle du fichier .DBF s'avèrent intéressantes pour indiquer si des données ajoutées sont à écrire :

```
IF DBFSIZE() - DBFDSKSIZE() > 2 * recsize()
COMMIT
ENDIF
```

La fonction CSETSAFETY(.T.) permet de basculer un commutateur interdisant d'écraser les fichiers.

☛ La gestion de l'«état» — status — (25 fonctions)

Il s'agit, là aussi, de fonctions maintenues pour raisons de compatibilité avec Nantucket Tools I, puisque Clipper 5 a la fonction SET() qui traite ce sujet. Il y a également les fonctions gérant l'état de CAPS NUM... : KSETINS(), KSETNUM(), KSETCAPS() et KSETSCROLL(). SETLASTKEY(«code inkey») permet de restaurer la «lastkey».

C'est utile, si, après débranchement, SET KEY ou VALID ayant éventuellement utilisé le clavier, on veut fixer ou restaurer la touche de sortie, pour la suite du programme.

☛ Le système — DOS — (24 fonctions)

Les fonctions BOOTCOLD() et BOOTWARM(), contrairement à ce que leur intitulé fait penser, ne font que bloquer l'ordinateur. Certaines fonctions procurent des codes d'erreurs DOS, d'un intérêt discutable.

EXENAME() donne le path et nom de l'EXE en cours.

☛ Divers (22 fonctions)

On retiendra :

```
KBDISABLE()
// inhibe le clavier y compris Ctrl Alt Del

KBDEMULATE()
// injection clavier au niveau Bios : utile
// pour la gestion de souris ou (en fin de
// programme) pour lancer la suite

KBDSTAT()
// état du clavier, y compris SHIFT, CTRL et ALT

SHOWKEY()
// affichage automatique de l'état du
//clavier avec textes libre

MILLISEC()
// attente délai

SETTIC()
// changement de la fréquence
// d'interruption timer,
// sans perturber l'heure

SOUND()
// musique

SPEED()
// vitesse du processeur par rapport au
// PC-XT
```

La fonction KEYSEC() permet d'injecter un code au clavier après un certain délai. Selon les paramètres d'appel, le processus peut se répéter automatiquement. Le délai peut être relancé si une frappe réelle se produit.

Ceci permet les démos automatiques, les «times out», l'affichage d'horloge, l'affichage d'avancement de traitement par interruption évitant de ralentir celui-ci, les COMMIT automatiques, etc.

☛ La gestion du réseau (44 fonctions)

Clipper fournit le minimum des fonctions de bloquage permettant la programmation d'applications pour réseaux locaux. C'est souvent suffisant. Les fonctions supplémentaires fournies dépendent presque toutes du type de réseau.

Les deux principaux types (et leurs compatibles) sont supportés :

- Novell : ELSx, NetWare 2.1 / 386,
- PC LAN / MS-NET (IBM et Microsoft).

Les fonctions générales communes comprennent le test d'une unité, pour savoir si elle appartient au serveur (NETDISK), et le test du type de réseau.

Les fonctions dédiées Novell autorisent la gestion de la capture des sorties imprimantes, ce qui est nécessaire pour les rediriger vers une imprimante du réseau. Ceci inclut le paramétrage (choix de la file, bannières, etc.).

Elles permettent d'appeler des fonctions de Novell de copie de fichiers sur le serveur, sans passer sur le

réseau. Il y a, enfin, les fonctions classiques relatives à l'utilisateur indiquant son numéro de station, son identification, son nom de log in, son numéro de carte réseau, sa place libre sur le serveur.

Nantucket Tools ne comporte pas de fonctions exploitant le système TTS (Transaction Tracking System) pour la sécurisation des fichiers. Les fonctions dédiées à PC LAN/MS-NET ne donnent pas accès à la redirection vers les unités du serveur ni au test de l'imprimante réseau.

☛ Les mathématiques (23 fonctions)

Il s'agit des fonctions trigonométriques, du log à base 10 et des fonctions financières classiques de calculs d'intérêts d'emprunts.

☛ Les accès directs en mémoire — peek/poke — (9 fonctions)

Il s'agit des célèbres PEEK et POKE, d'accès direct mémoire. Les fonctions de la librairie couvrent en fait la plupart des nécessités.

PEEKSTR() lit une chaîne entière sans boucler — donc rapidement.

☛ La saisie — GET/READ — (11 fonctions)

Rien d'intéressant ici, en Clipper 5, puisque les fonctions de ce groupe, maintenues par compatibilité permettaient les READ imbriqués en Summer 87. Elles sont d'ailleurs fournies en source Clipper 5. On regardera avec intérêt GETSECRET() pour saisir les mots de passe avec affichage d'étoiles.

En conclusion

Nantucket Tools II constitue un bon produit, très complet — mode graphique et souris mis à part.

Dommage que cette dernière ne soit pas traitée, car s'il est admissible d'acquérir une autre librairie pour le graphisme, les fonctions de souris s'avèrent très faciles à réaliser. L'autre reproche que l'on peut faire concerne l'absence d'une liste unique de fonctions dans les aides Norton — il faut toujours retrouver le chapitre, ce qui n'apparaît pas toujours évident.

Il est donc nécessaire d'avoir en main le Quick Reference Guide fourni avec Nantucket Tools. Un très gros avantage de ce produit est d'être supporté par Nantucket, ce qui représente un gage de pérennité et de sécurité pour toutes les fonctions liées aux caractéristiques internes de Clipper. ❐

Editeur : Nantucket
Distributeur : PC Technologie
Prix : 6 950 F HT

LES SEMINAIRES
ETAT DE L'ART

Micro & Communications

21-23 SEPTEMBRE OU 1-3 DECEMBRE 1992
Micro: la synthèse 1992
Nouveaux choix techniques et stratégiques, perspectives à 3 ans.

23-25 NOVEMBRE 1992 OU 22-24 MARS 1993
Réseaux Locaux
Architectures et alternatives techniques, panorama de l'offre commerciale.

21-23 SEPTEMBRE OU 23-25 NOVEMBRE 1992
Unix
Les connaissances indispensables : la synthèse en 3 jours.

15-16 SEPTEMBRE 1992 OU 2-3 FEVRIER 1993
Gérer la Micro
PC Management : organisation, méthodes, outils, démarches pratiques.

16-18 NOVEMBRE 1992 OU 17-19 MARS 1993
Bureautique
Nouveaux choix techniques, nouvelles stratégies pour les 5 prochaines années.

24-25 SEPTEMBRE 1992 OU 1-2 AVRIL 1993
La Sécurité Micro
De l'analyse du risque à la mise en place des parades : outils, méthodes, organisation.

26-27 NOVEMBRE 1992 OU 28-29 JANVIER 1993
Applications Client-Serveur
Principes, techniques, outils disponibles, démarche pratique de mise en œuvre.

26-27 NOVEMBRE 1992 OU 28-29 JANVIER 1993
Maintenance d'un Parc Informatique
Gestion de la maintenance: par qui, comment, à quel prix ?

17-18 SEPTEMBRE OU 26-27 NOVEMBRE 1992
Connexion de Micros
Connexion aux mainframes, réseaux locaux de micros : l'état de l'art et offre commerciale.

17-18 SEPTEMBRE 1992 OU 28-29 JANVIER 1993
Migrer vers les Systèmes Ouverts ?
Avantages et risques, choix techniques, démarches pratiques.

Programmes détaillés et inscriptions : Véronique Groud, CAP SESA INSTITUT - 207, rue de Bercy, 75012 Paris Tél. : 44.74.24.10

CAP SESA INSTITUT
L'Etat de l'Art

Service lecteurs n° 26 (dernière page)

PRATIQUE

TURBO VISION

Création d'un programme de changement de répertoire

TOM SWAN

La commande FileChange du nouvel IDE (interface de développement) du Turbo Pascal 6.0 (TP6) permet d'accéder directement à un sous-répertoire. Après avoir lancé cette commande, on utilise les touches de déplacement du curseur ou la souris pour sélectionner le nom de répertoire.

Mais ce n'est guère pratique de charger TP6 pour un simple changement de répertoire. Le programme proposé, NAV.PAS écrit sous Turbo Vision, exécute l'équivalent de cette commande.

L'IDE du TP6 étant écrit en Turbo Vision —nom désignant la série des outils pour les applications orientées objet qui accompagne le TP6— , il est facile d'inclure des propriétés de l'IDE aux programmes écrits en Turbo Pascal.

Fonctionnement de NAV.PAS

La directive {$X+} de la première ligne du listing appelle la nouvelle option Syntaxe étendue du TP6. Lorsque cette directive est active, les fonctions peuvent être appelées comme des déclarations.

Par exemple, si MaFonction constitue une fonction retournant une valeur entière à x, variable entière, MaFonction devrait normalement être appelée par :

```
x = MaFonction;
```

Avec la syntaxe étendue de TP6 activée, l'appel se fait simplement par :

```
MaFonction
```

MaFonction retourne toujours un entier, mais lorsqu'elle est appelée comme une procédure, le résultat de la fonction reste ignoré. Le programme n'a plus besoin de sauvegarder une variable inutile.

La procédure ChangeDir représente un autre exemple de syntaxe étendue.

```
procedure ChangeDir;
var
D PChDirDialog;
begin
      D=New(PChDirDialog,Init(cdNormal,101));
      if ValidView(D)<>nil then begin
      DeskTop^.ExecView(D);
      Dispose(D, Done);
end
end;
```

Après avoir créé le dialogue Change Directory (changer de répertoire) et lui avoir assigné l'adresse D, une déclaration If vérifie si le dialogue View est valide. Le contraire semble peu probable dans un programme d'une taille aussi réduite, mais cela ne coûte rien de vérifier.

Si View est valide, le programme appelle DeskTop^.ExecView(D), qui traite toutes les commandes de changement de répertoire décrites plus haut.

ExecView est une fonction qui retourne un mot indiquant le statut de la vue exécutée. Dans notre cas, le résultat n'a aucun sens.

Le fait d'appeler une fonction comme une procédure évite le pas intermédiaire de mémoriser cette valeur dans une variable temporaire.

La procédure principale de NAV, Handle Event, dont dépend ChangeDir, appelle la procédure TApplication.HandleEvent pour traiter la plupart des événements —sollicitation des boutons de la souris, touches enfoncées... — produits lors de l'exécution du programme.

```
procedure TNavApp.HandleEvent;
begin
TApplication.HandleEvent(Event);
if Event.What = evCommand then begin
if Event.Command = cmChangeDir
then ChangeDir
```

La commande CD du DOS devient fastidieuse à employer lorsque le nombre de répertoires d'un disque dur dépasse la centaine. Naviguer d'un répertoire à l'autre entraîne alors un ralentissement comparable à la circulation dans une grande ville aux heures de pointe. D'où l'intérêt d'un programme qui conduit directement au répertoire sélectionné et que nous allons réaliser en Turbo Pascal 6.0 et en Turbo Vision.

```
else Exit;
end;
end;
```

Cette importante étape effectuée, les commandes qui correspondent au programme sont détectables. Si un événement de type commande est égal à la constante cmChangeDir, le programme appelle ChangeDir pour commencer le dialogue de changement de répertoire. La constante cmChangeDir est déclarée au début de NAV.PAS.

Les procédures InitMenuBar et InitStatusLine se révèlent analogues à divers programmes de démonstration fournis dans Turbo Vision. Ces procédures préparent la barre de menus et la ligne d'état affichée en haut et en bas de l'écran.

La commande ChangeDir se trouve incluse dans la barre de menus afin de permettre à l'utilisateur de réouvrir le dialogue de changement de répertoire dans le cas où il aurait fermé cette fenêtre par erreur. Le programme principal de NAV s'avère un peu plus long que d'habitude.

```
var
NavApp TNavApp; NavEvent; TEvent;
begin
NavApp.Init;
NavEvent.what = evCommand;
NavEvent.Command = cmChangeDir;
NavEvent.InfoPtr = nil;
NavApp.PutEvent(NavEvent);
NAvApp.Run;
NavApp.Done;
end
```

La plupart des applications en Turbo Vision n'appliquent que trois méthodes aux objets TApplication global, ou dérivés : Init, Run et Done. Dans NAV, TNavApp constitue l'objet dérivé, déclaré vers le début du listing comme procédant de TApplication. Tous les programmes Turbo Vision demandent un objet analogue ainsi que les méthodes de remplacement pour les trois procédures virtuelles énumérées ci-dessus. Appeler les méthodes Init, Run et Done pour une exécution de TNavApp initialise, exécute et remet à 0 l'application Turbo Vision utilisant ces procédures de remplacement.

NavEvent est un objet TEvent. Fixer ses champs d'application comme le fait le programme principal simule la sélection de la commande ChangeDir.

```
NavEvent.what = evCommand;
NavEvent.Command = cmChangeDir;
NavEvent.InfoPtr = nil;
```

Après avoir fixé les valeurs des champs, le programme appelle PutEvent pour l'objet NavEvent, plaçant ainsi l'événement simulé dans la queue des commandes.

En conséquence, NAV considère que quelqu'un a choisi la commande ChangeDir, entraînant par HandleEvent et ChangeDir l'ouverture du dialogue de changement de répertoire au début du programme.

Compilation et exécution de NAV

Pour exécuter le programme sous DOS, lancer NAV et pointer le nom du répertoire choisi dans la fenêtre ChangeRépertoire à l'aide du clavier ou de la souris. Le changement de champ se fait par TAB au clavier, ou avec la souris.

Cliquer sur le bouton OK ou utiliser la touche K pour confirmer le répertoire choisi, ce qui ferme également la fenêtre, laquelle peut être réouverte à l'aide de F3.

Cliquer sur Revert ou employer R pour revenir au point de départ.

Cliquer sur le bouton de fermeture du dialogue ou appuyer sur Echap pour sortir sans changer de répertoire. Sortir par ALT-X. ❐

Voulez-vous faire ce qui vous plait et gagner 50.000 francs HT par mois ?

Gagnez 3.000 francs/jour et plus

Savez-vous qu'un indépendant facture en moyenne 3.000 francs (hors taxes) par jour ? Que certains spécialistes gagnent 8.000 francs HT par jour ? Et vous, connaissez-vous votre véritable valeur ?

Soif de liberté ?

Vous ne supportez plus le métro-boulot-dodo ni les 39 heures par semaine ? Vous aimez travailler la nuit ou le dimanche ? Le statut d'indépendant est fait pour vous. Chez vous ou chez vos clients. Cinq jours par semaine ou 3 demi-journées par mois. A la campagne ou à la plage. Vous travaillez comme vous le décidez.

Toutes compétences bienvenues

Développeur ou formateur, assistant micro ou consultant réseaux, autodidacte ou ingénieur diplômé, vous maîtrisez l'asm 80x86, le C, le vidéotex, des produits bureautiques ou même le Basic... des milliers d'entreprises ont besoin de vous ! Ne vous bradez pas. Vendez-vous à votre véritable valeur.

Un marché énorme

De plus en plus d'entreprises font appel à des indépendants. Moins coûteux, plus flexible et beaucoup plus simple, ce statut cumule tous les avantages. Ce n'est donc pas un hasard si le marché double tous les deux ans.

Moins de contraintes

Avec ce nouveau statut, vous serez enfin maître de votre temps. Pour pourrez consacrer du temps à votre famille, vos loisirs et vous organiser pour éviter la foule des week-end. Et puis pourquoi pas, partir en vacances au bout du monde sur un coup de tête. Sans avoir de compte à rendre à qui que ce soit !

Vivez HT et déduisez tout !

Indépendant, vous récupérez la TVA de toutes vos dépenses professionnelles. Déduisez de vos revenus votre matériel informatique, votre loyer, vos frais de téléphone et de voiture... Arrangez-vous pour trouver des missions à l'étranger et voyager à l'oeil !

Accélérateur phénoménal

Voulez-vous accumuler les expériences et gonfler votre C.V. hyper rapidement ? Voulez-vous 'visiter' 15 sociétés en 2 mois ? Voulez-vous gagner des années dans votre plan de carrière ? Voulez-vous faire exploser votre carnet d'adresses ? Le statut d'indépendant est un formidable accélérateur de votre carrière professionnelle. D'après un cabinet de recrutement une année en indépendant en vaut trois en salarié !

Comment ?

Choisissez le bon statut ! Devenez indépendant. Dîtes non au salariat -véritable esclavage des temps modernes- et à ses charges sociales exorbitantes. Dîtes oui, à la liberté de travail et à l'indépendance financière. Choisissez vos missions et vos lieux de travail. Vendez-vous à votre véritable valeur et ne faites plus les choux gras des société de service... Le "Manuel de l'Indépendant en Informatique" vous dit tout. Véritable guide initiatique, il a été écrit par des indépendants puis enrichi par des spécialistes des domaines fiscaux et sociaux. Tout ce que vous devez savoir est exposé en détail dans ce petit ouvrage de 200 pages :

- Pourquoi être indépendant ?
- Côté social (cotisations et couvertures)
- Côté fiscal (obligations et impôts)
- Comment trouver vos clients ?
- Conseils au débutant et trucs des professionnels
- Des premiers contacts aux réseaux de relations...
- Comment et combien facturer ?
- Quand et comment vous déclarer ?
- Tous les pièges à éviter
- Toutes les adresses indispensables
- Et des tableaux récapitulatifs concis et à jour...

Des astuces ... enrichissantes

Vous y trouverez tous les trucs que seule l'expérience permet de connaître. Ceux que jamais votre inspecteur des impôts ne vous dévoilera. Comment éviter les contrôles fiscaux ? Comment bénéficier d'un abattement de 20% sur votre bénéfice ? Toutes les adresses pour que les contrats vous tombent tout cuits dans les mains. Comment vous lancer sans aucun risque ? Comment toujours bénéficier d'une bonne avance de trésorerie ? Comment passer par un cabinet comptable sans que cela ne vous coûte un centime ? Etc.

Les premiers lecteurs emballés

"génial votre bouquin (...) le côté juridique et fiscal est accessible et complet et pourtant votre manuel n'a rien à voir avec les bouquins juridiques et fiscaux que l'on trouve sur le marché (...) merci." Gilles C. Paris XV. Ancien salarié dans une SSII.
"mais pourquoi avez-vous attendu si longtemps ? Cela fait cinq ans que je cherchais un livre comme celui-ci (...) rien ne manque" Frédéric T. Lyon
"pour moi qui suis étudiant, c'est la solution idéale pour vendre mes compétences tout en continuant mes études" Frédéric D. Paris

Pourquoi cette offre ?

Parce que l'union fait la force. Plus nous serons nombreux, plus nous serons crédibles. Faire appel à un indépendant deviendra aussi courant que de confier la rédaction de ses contrats à un avocat. Dix fois plus d'indépendants outre-Alantique, la vague va t-elle déferler en Europe ?

Qui sommes-nous ?

L'Association des Indépendants en Informatique (A.I.I.). Créée par un petit groupe d'indépendants au mois de janvier 92. Notre but : ôter à ce statut son côté marginal et le faire reconnaître comme un métier à part entière, au même titre que celui d'avocat ou de médecin. Et permettre à chacun de vendre ses compétences sans intermédiaire au client final... Pour cela, il nous faut réunir des forces -et des fonds- pour mener une véritable campagne d'information -prévue pour début 93- sur notre statut.

Assistance gratuite

N'hésitez pas à nous contacter pour avoir de plus amples renseignements. Appelez-nous au (1) 47.45.02.57 ou posez-nous vos questions directement par minitel sur le 3617 code AII. Nous y répondrons dans les plus brefs délais.

Réponse rapide conseillée

Nous serons heureux de joindre à votre manuel le pin's de l'AII (tirage limité à 1000 exemplaires). Deux couleurs sur fond noir, sobre et chic ! Mais attention, cette offre est uniquement valable dans la limite des stocks disponibles.

Satisfait ou remboursé

Si pour quelque raison que ce soit vous n'étiez pas entièrement satisfait de votre ouvrage, il vous suffirait de nous le renvoyer dans un délai de 30 jours (à compter de la date de réception) pour être intégralement remboursé. Ceci, sans que vous ayez à nous fournir le moindre justificatif.

Bon d'essai GRATUIT et sans risque

à découper (ou photocopier) et retourner à

Association des Indépendants en Informatique

5, rue Charcot - 92200 Neuilly

❑ Oui, je veux tout savoir sur le statut d'indépendant. Envoyez moi votre manuel par retour du courrier. J'ai bien noté que si je ne suis pas entièrement satisfait, je pourrai, dans un délai de 30 jours à compter de la date de réception, vous le renvoyer pour être intégralement remboursé. Ceci, sans avoir à vous fournir le moindre justificatif.

Sous cette garantie totale, je vous règle par ❑ chèque ❑ mandat :

❑ 249 francs (commande franco de port expédiée au tarif lent)

❑ 269 francs (envoi en Colissimo ou Etranger et DOM/TOM)

Je préfère un envoi en contre remboursement :

❑ je règlerai alors au facteur la somme de 299 francs (50 francs de supplément)

Mes coordonnées:

❑ Madame ❑ Monsieur ❑ Mademoiselle

Nom: _____ Prénom: _____

Adresse: _____

Code postal: _____ Ville: _____

Une facture justificative avec TVA vous sera adressée avec votre manuel - Offre valable dans la limite des stocks disponibles

Service lecteurs n° 27 (dernière page)

TURBO VISION

FIGURE NAV.PAS

```
FICHIER: 1.NAV.PAS
{$X+}
program NAVIGUER; (*par Tom Swan*)
uses
DOS,Objects,Drivers,Views,Menus,App,StdDlg;
const
   cmChangeDir = 1000;
type
   TNavApp = object(TApplication)
      procedure  Handle Event(var Event:
      TEvent);virtual;
      procedure InitMenuBar; virtual;
      procedure InitStatusLine; virtual
   end;
procedure TNavApp.HandleEvent(var
Event:TEvent);
procedure ChangeDir;
var
   D: PChDirDialog;
begin
   D:= New(PChDirDialog, Init(cdNormal, 101));
   if ValidView(D)<>nil then begin
      DeskTop^.ExecView(D);
      Dispose(D, Done);
   end
end;
begin
   TApplication.HandleEvent(Event);
   if Event.What = evCommand then begin
      if Event.Command = cmChangeDir
      then ChangeDir
      else Exit;
   end;
end;
procedure TNavApp.InitMenuBar;
var
   R: TRect;
begin
R.B.Y : =R.A.Y + 1;
   GetExtent(R);
      MenuBar := NEw(PMenuBar,Init(R, NewMenu(
      NewSubMenu('Menu', hcNoContext, NewMenu(
    NewItem('ChangeDir','F3',kbF3,cmChangeDir,
    hcNoContext,NewItem('Exit','AltX',kbA
    ltX, cmQuit,hcNoContext,nil))),nil))));
end;
procedure TNAvApp.InitStatusLine;
var
   R: TRect;
begin
   GetExtent(R);
   R.A.Y := R.B.Y -1;
   StatusLine := New(PStatusLine, Init(R,
   NewStatusDef(0,$FFFF,
      NewStatusKey(",kbF10,cmMenu,
      NewStatusKey('Alt-Exit',kbAltX,cmQuit,
      NewStatusKey('F3 Change Dir,
      kbF3,cmChaqngeDir, nil))), nil))));
end;

var
   NavApp: TNavApp;  NavEvent; TEvent;
begin
   NavApp.Init;
   NavEvent.what := evCommand;
   NavEvent.Command := cmChangeDir;
   NavEvent.InfoPtr := nil;
   NavApp.PutEvent(NavEvent);
   NAvApp.Run;
   NavApp.Done;
end
```

WIN - DOS - OS2 - UNIX - MOTIF

Micro Sigma

Outils de Développement

POO Portable

Nous vous proposons la famille de classes C++ de Glockenspiel et de RogueWave ainsi que notre générateur d'applications **CommonView.**

CommonView - l'Interface Graphique Utilisateur sous formes de classes portables Windows, PM, et MOTIF.

CommonBase - Librairie de classes d'Interface sur base de données ORACLE, Informix, SyBase, SQL Server. Portable entre MS-Windows, PM, et MOTIF.

AppDesigner.C++ - Création interactive de l'interface graphique. Génère le code C++ pour CommonView

Tools.H++ et **Math.H++** - librairies de classes générales et de calcul matriciel de Rogue Wave.

Le Mode Protégé ou comment dépasser les 640K

HIGH C de Metaware - Un standard de qualité, beaucoup employé par les développeurs C sous AUTOCAD.

WATCOM C et FORTRAN Version 9.0 - peut-être le meilleur compilateur C mode protégé disponible aujourd'hui.

Microway 386 NDP FORTRAN, C et C++ - Une famille de compilateusr qui a fait ses preuves.

LAHEY Fortran EM32 - Une réfèrence en matière de Fortran. Disponibles avec de nombreux outils annexes.

386 DosExtender Toolkit de **Pharlap** - Toute la puissance des 386/486 accessible sous DOS par le biais de ce toolki.

Adressez vous à une entreprise compétente pour vos outils de développement pour MS-DOS, MS-Windows, OS2/PM, UNIX, AIX, SUN, DEC Station, HP9000, VAX

Les noms évoqués sont des marques déposées d'IBM, Oracle, Informix, Glockenspiel, Microsoft, Gupta,Autodesk, Digital Equipment et RogueWave.

FAXFACTS

Ce service de documentation par télécopie vous permet d'obtenir directement depuis votre FAX une documentation sur la plupart des produits disponibles chez Micro Sigma. Pour cela appelez le **47.63.59.93** et suivez les instructions. La documentation NO. 1 contient la liste des documentations disponibles sous **FAXFACTS.**

Les Services MICRO SIGMA

Véritable spécialiste des outils de développement, nous avons limité notre choix afin de vous offrir un **Service Professionnel**.

- Conseil avant acquisition
- Support Technique
- Mises à Jour
- Contrats de Maintenance
- Formation **CommonView**

Du fait que nous importons tous nos produits directement, nous bénéficions de rapports privilégiés avec les concepteurs de ces outils.

Demande de Documentation

Si vous désirez plus d'informations sur les produits ci dessus, vous pouvez nous demander notre catalogue, consulter notre service **FAXFACTS,** ou nous renvoyer ce coupon réponse après avoir coché les produits qui vous intéressent.

☐ Mode Protégé	☐ POO Portable	☐ ProtoView
☐ Dév C AUTOCAD	☐ CommonView	☐ Programmation Linéaire
☐ Windows 32 bits	☐ Librairies Graphiques	☐ Cross Dév - 80x86
☐ Panel Plus	☐ Cross Développement	☐ Emulateurs
☐ Noyaux Temps Réel	☐ Assurance Qualité	☐ Clés de Protection
☐ C Scientific Library	☐ Traducteurs Fortran-C	☐ **CATALOGUE**

Micro Sigma, S. A.
42, rue Legendre,
75017 - Paris.
TEL : 46.22.99.88
FAX : 47.63.84.42

Nom_____
Service_____
Entreprise_____
Adr._____
Adr._____
Code_____ Ville_____

THÉORIE

LES INDEX DE FICHIERS À BASE DE B-ARBRES

Accéder rapidement aux données

ERIC DE KERVALEC

L'intérêt des gestionnaires de fichiers tient à leur capacité à retrouver de manière efficace les données répondant à certains critères (par exemple, tous les clients habitant telles villes et dont les commandes dépassent un certain montant). Avec l'augmentation régulière des capacités de stockage des périphériques d'ordinateurs (disques durs, CD-ROM, etc.), les données à extraire peuvent être stockées dans des fichiers de très grande taille. L'algorithme de recherche utilisé joue donc un rôle essentiel dans la performance globale du système.

La méthode habituelle pour faciliter ce type de recherche consiste à construire un index qui est automatiquement mis à jour lors de chaque modification d'un enregistrement. Cet index est généralement stocké dans un fichier séparé de celui qui contient les enregistrements proprement dits (le programme Btrieve de Novell constitue toutefois une exception notable à cette règle). Les enregistrements eux-mêmes sont ajoutés au fichier principal dans l'ordre de leur saisie (en réutilisant éventuellement la place laissée vacante par les enregistrements marqués comme étant «détruits» du fichier), tandis que le fichier d'index ne contient que les clés sur lesquelles se feront les recherches ultérieures (les clés ne représentent habituellement qu'une fraction des données stockées dans un enregistrement). Ces clés doivent donc être organisées de manière à pouvoir retrouver rapidement l'adresse de l'enregistrement associé à une clé donnée (cette adresse peut représenter le numéro de l'enregistrement ou le déplacement correspondant dans le fichier principal).

Les contraintes

Dès que l'on aborde la gestion de fichiers de grande taille, le facteur limitatif devient l'accès au périphérique de stockage, autrement dit, la lecture ou l'écriture d'un secteur sur le disque dur (ou sur le CD-ROM). Les méthodes d'indexation tiennent compte de cette contrainte et cherchent à minimiser le nombre de secteurs à lire sur le disque pour localiser un enregistrement donné. D'une manière générale, on mesure les performances d'un algorithme d'indexation selon quatre critères :

- le temps de recherche : il correspond au temps nécessaire pour parcourir l'index et obtenir l'adresse de l'enregistrement recherché;
- le temps d'insertion : il mesure le temps nécessaire à l'ajout d'une nouvelle valeur dans le fichier d'index. Chaque insertion impose généralement un parcours de l'index pour déterminer l'emplacement où la nouvelle valeur sera insérée. Il peut également s'avérer utile de procéder à une réorganisation partielle de la structure de l'index;
- le temps de destruction : il se mesure de manière similaire au temps d'insertion. Il faut parcourir l'index pour trouver la valeur à détruire et réorganiser éventuellement l'index.
- l'espace occupé par l'index : il doit évidemment être le plus faible possible. Outre l'économie réalisée sur la capacité de stockage du périphérique, un index compact permet de conserver en mémoire un plus grand nombre de clés, ce qui augmente la performance globale du système.

Les algorithmes d'indexation à base de B-arbres étudiés ici obtiennent de bons résultats sur l'ensemble des critères précités. D'autres méthodes d'indexation peuvent se révéler mieux adaptées à certains cas particuliers (par exemple, une base de données sur CD-ROM dont le contenu n'est jamais modifié). La souplesse d'utilisation des B-arbres leur permet toutefois de couvrir la grande majorité des besoins en matière d'indexation, ce qui explique sans doute pourquoi de nombreux SGBD du commerce gèrent leurs index à l'aide d'une méthode dérivée des B-arbres. Parmi les plus connus, on peut notamment citer FoxPro 2.0 (Fox Software), Btrieve (Novell), Paradox (Borland) et db_Vista (Raima Corporation, distribué en France par Ise-Cegos).

Les index sont utilisés pour accéder rapidement aux données contenues dans des fichiers de grande taille. Parmi les diverses méthodes existantes, les algorithmes à base de B-arbres sont aujourd'hui très populaires en raison de leurs performances et de leur souplesse d'utilisation.

Comment un arbre accélère-t-il la recherche de l'enregistrement correspondant dont la clé a une valeur donnée ? Pour répondre à cette question, considérons tout d'abord un type d'arbre particulier appelé «arbre binaire». Il est important d'en comprendre le fonctionnement, car les B-arbres étudiés plus loin procèdent des mêmes principes.

Un arbre binaire se caractérise par le fait que chaque nœud a deux enfants au maximum. L'arbre est ordonné de manière à ce que toutes les clés de valeur inférieure à celle du nœud courant soient situées sur la branche de gauche de ce dernier, tandis que la branche de droite mène aux nœuds de valeur supérieure ou égale à celle du nœud courant. La figure 2 représente un arbre binaire construit selon ces conventions. Voici comment s'effectue dans cet arbre la recherche de l'enregistrement associé à la clé de valeur 16 : la valeur recherchée étant inférieure à celle de la racine (25), il faut donc suivre la branche de gauche vers le nœud de valeur 15; puisque 16 est supérieur à 15, on suit la branche de droite vers le nœud de valeur 18; comme 16 est inférieur à 18, on suit alors la branche de gauche qui mène à la valeur recherchée. Il ne reste plus alors qu'à lire dans ce nœud l'adresse de l'enregistrement correspondant à cette clé qui est stocké dans le fichier principal de la base de données.

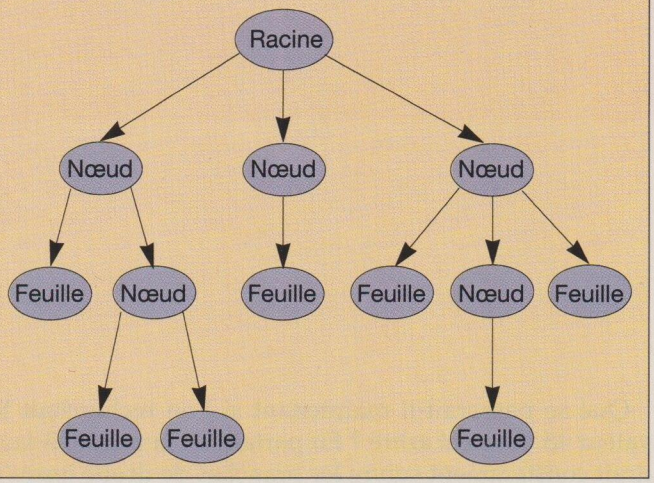
L'arbre binaire représenté sur la figure 2 contient 13 clés et a une profondeur de 4 niveaux, puisqu'il faut traverser au maximum 4 branches pour aboutir au nœud le plus éloigné de la racine. Une recherche dans cet arbre nécessite donc la lecture de 4 nœuds au maximum (si l'on suppose que la racine est conservée en mémoire). A titre de comparaison, une recherche séquentielle dans la même liste de valeurs nécessiterait la lecture de 6 ou 7 nœuds en moyenne, et jusqu'à 13 nœuds dans le pire des cas. L'arbre binaire offre donc déjà de bien meilleures performances qu'une recherche séquentielle, et l'écart augmente (en faveur de la recherche par arbre, bien sûr) avec le nombre de valeurs stockées (on suppose que la lecture d'un nœud impose à chaque fois un accès au disque).

Que se passerait-il maintenant si l'on recherchait la valeur 45 dans cet arbre ? En partant de la racine, il faudrait constamment suivre les branches de droite jusqu'à arriver au nœud de valeur 50. Puisque 45 est inférieur à 50, le nœud en question se trouverait sur la branche de gauche à partir du nœud de valeur 50, mais cette branche n'existe pas ! La valeur recherchée n'est donc pas présente dans l'arbre binaire. Cette recherche indique toutefois l'endroit où il faudrait insérer la valeur en question dans l'arbre, s'il fallait lui ajouter la clé de valeur 45.

L'ARBRE INFORMATIQUE : EXPLICATIONS

Pour comprendre les explications relatives à la recherche d'une valeur dans un arbre, il est nécessaire de définir quelques termes qui concernent tous les types d'arbres informatiques (dont les arbres binaires et les B-arbres ne représentent que quelques cas particuliers). En informatique, un arbre organise les données selon une structure hiérarchique. Le nom d'«arbre» vient du fait que le dessin de cette structure ressemble à un arbre «naturel» renversé. Les endroits où les «branches» se rejoignent sont appelés des «nœuds». Le nœud situé au niveau le plus élevé est la «racine» de l'arbre, tandis que les «feuilles» constituent le niveau inférieur. Les autres nœuds de l'arbre sont nommés les «nœuds intérieurs» ou simplement les «noeuds». On utilise généralement des termes empruntés à la généalogie pour désigner les relations qui existent entre les divers nœuds d'un arbre. Chaque nœud en dehors de la racine a exactement un «père» (le nœud de niveau supérieur auquel il est relié par une branche) et peut avoir plusieurs «enfants» ou «descendants». Plusieurs nœuds qui partageant le même père sont des «frères», et ainsi de suite. La figure 1 montre comment ces termes s'appliquent à un arbre simple. La «profondeur» de l'arbre représenté sur cette figure est de trois niveaux, car il faut parcourir au maximum trois branches pour aller de la racine à la feuille la plus éloignée.

FIGURE 1 : *un arbre informatique*



Bien que la méthode d'insertion décrite ci-dessus soit simple à comprendre, elle présente un risque important de dégénérescence lorsque les clés sont ajoutées de manière presque ordonnée (un cas fréquent dans la pratique).

La figure 3 représente l'arbre binaire obtenu si l'on part d'un arbre vide auquel on ajoute successivement les valeurs 15, 16, 18, 21, 25, 5, 32, 41, 40, 43, 50, 19 et 23. Cet arbre contient les mêmes valeurs que celui de la figure 2, mais sa profondeur est de 8 niveaux. Dans ce cas, une recherche peut donc obliger à lire jusqu'à 8 nœuds, ce qui n'est guère plus avantageux que la simple recherche séquentielle.

Il serait évidemment possible de réorganiser l'arbre pour obtenir une structure plus équilibrée, mais cette solution nécessiterait un temps de traitement important pendant lequel l'index ne pourrait être utilisé pour accéder au fichier de données.

Equilibrer les arbres

Pour éviter d'aboutir à la situation précédente, le plus simple est d'adopter une méthode d'insertion qui maintienne l'arbre binaire en «équilibre».

Cette propriété signifie que, pour n'importe quel nœud, la profondeur du sous-arbre de gauche diffère de celle du sous-arbre de droite d'au plus un niveau. Elle garantit également un bon «remplissage» de chaque niveau, ce qui réduit le temps de recherche moyen. A titre d'exemple, l'arbre binaire dessiné sur la figure 4 est équilibré, et sa profondeur n'est plus que de 3 niveaux (bien qu'il contienne les mêmes valeurs que les arbres des figures 2 et 3).

L'utilisation d'un arbre binaire équilibré suppose toutefois que l'on effectue quelques réorganisations pour conserver l'équilibre de l'arbre après chaque insertion ou destruction. Les algorithmes employés pour obtenir ce résultat sont similaires à ceux décrits ci-après dans le cadre des B-arbres : ils ne sont donc pas détaillés dans le cas particulier des arbres binaires.

Les B-arbres

Les B-arbres généralisent les notions étudiées ci-dessus sur les arbres binaires équilibrés en permettant le stockage de plusieurs clés dans chaque nœud de l'arbre.

Un nœud qui contient N clés possède exactement N+1 enfants, ce qui augmente le «facteur de branchement» (fan-out) des B-arbres par rapport aux arbres binaires. Pour un même nombre de clés, la profondeur d'un B-arbre est donc plus faible que celle de l'arbre binaire équilibré correspondant.

On appelle l'ordre d'un B-arbre le nombre maximal de clés que peut contenir un nœud, augmenté d'une unité (l'ordre d'un B-arbre correspond donc au nombre maximal d'enfants de chaque nœud). Un B-arbre d'ordre M doit satisfaire les trois propriétés suivantes :

1) La racine de l'arbre possède au minimum deux enfants, sauf si l'arbre est réduit à sa racine.

2) Chaque nœud possède au maximum M enfants et chaque nœud autre que la racine est rempli au moins à la moitié de sa capacité maximale. Cela signifie que tous les nœuds autres que la racine et les feuilles de l'arbre ont entre M/2 et M enfants (l'arrondi dans

FIGURE 2 : *un arbre binaire ordonné*

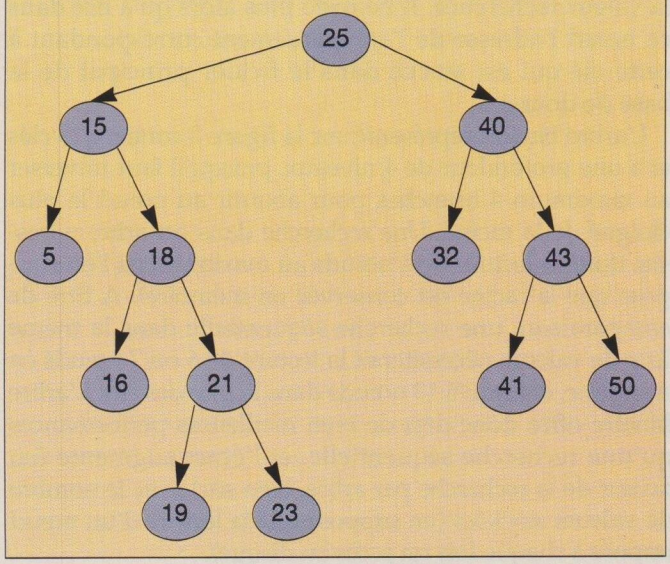
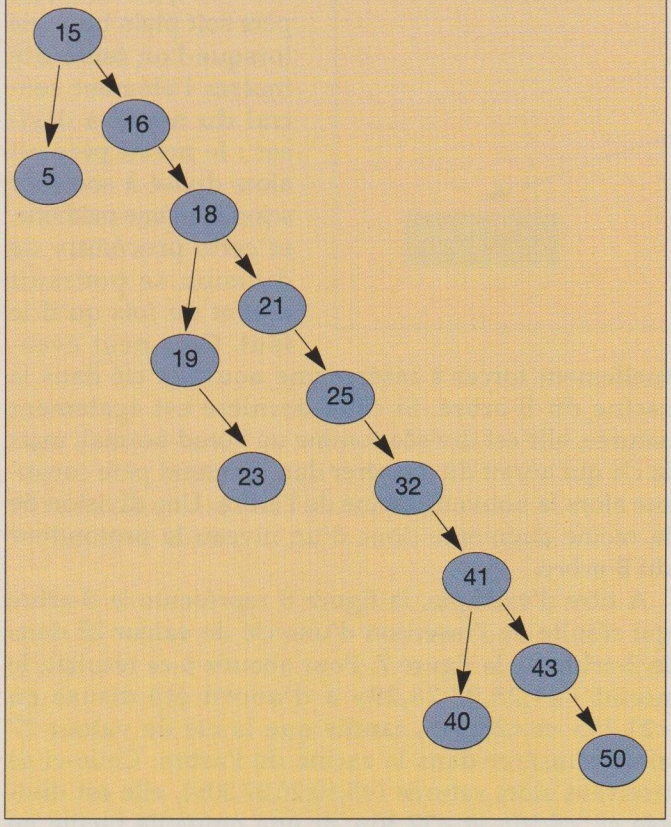


FIGURE 3 : *un arbre binaire dégénéré*



l'expression M/2 a lieu vers l'entier supérieur le plus proche : chaque nœud d'un B-arbre d'ordre 5 possède au minimum 3 enfants). En raison de la relation qui lie le nombre de clés que contient un nœud et celui de ses enfants, cette règle impose également à chaque nœud un seuil minimal de «remplissage» en nombre de clés (chaque nœud d'un B-arbre d'ordre 5 doit ainsi contenir au moins 2 clés).

3) Un B-arbre est parfaitement équilibré : toutes les feuilles sont à la même profondeur par rapport à la racine. Cette propriété garantit un bon niveau de performance quelle que soit la position de la clé recherchée dans le B-arbre.

La figure 5 représente un B-arbre d'ordre 5. La profondeur de cet arbre est seulement d'un niveau, bien qu'il contienne 15 clés (soit deux clés de plus que l'arbre binaire équilibré de la figure 4 qui s'étendait, quant à lui, sur 3 niveaux).

Un nœud occupe généralement un secteur sur le disque dur. La profondeur de l'arbre correspond donc au nombre maximal d'accès disque nécessaire pour localiser l'enregistrement associé à une clé donnée.

On peut prouver mathématiquement qu'un B-arbre d'ordre M qui contient K clés a une profondeur inférieure ou égale à $(1+(\log((K+1)/2)/\log((M/2)+1))$, ce qui permet de calculer le temps maximal de recherche d'une clé (un arbre binaire correspond au cas particulier où M=2).

Les méthodes d'accès à une branche

Par suite des clés multiples contenues dans chaque nœud, le parcours d'un B-arbre est plus complexe que celui d'un arbre binaire. Deux règles permettent de déterminer quelle branche suivre lors d'une recherche : à l'intérieur de chaque nœud, toutes les clés sont classées par ordre ascendant; pour chaque clé de valeur k, la branche de gauche qui part de cette clé contient les clés de valeur inférieure à k, tandis que la branche de droite contient les clés de valeur supérieure ou égale à k.

Lors de la recherche d'une clé, il faut donc appliquer pour chaque nœud (y compris la racine) la règle suivante : si la clé recherchée est inférieure à la première clé contenue dans le nœud, prendre la branche la plus à gauche. Si la clé recherchée est comprise entre la première et la deuxième clé du nœud, suivre la deuxième branche à partir de la gauche (et ainsi de suite). Si la clé recherchée est supérieure à la dernière clé du nœud (celle qui a la plus grande valeur), prendre la branche la plus à droite. Répéter ensuite cette opération à chaque nœud jusqu'à l'obtention de la valeur recherchée (ou de l'endroit où elle devrait se trouver si la clé correspondante n'existe pas dans l'arbre).

Les opérations d'insertion et de destruction de clés

FIGURE 4 : *un arbre binaire équilibré*

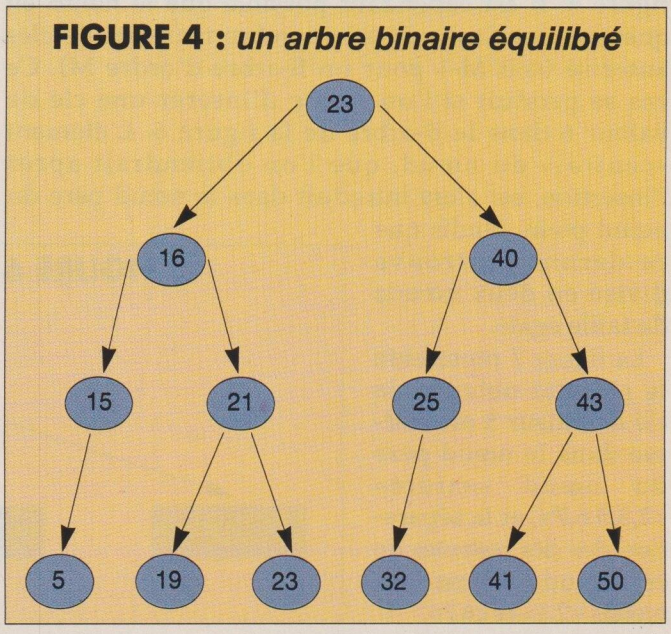
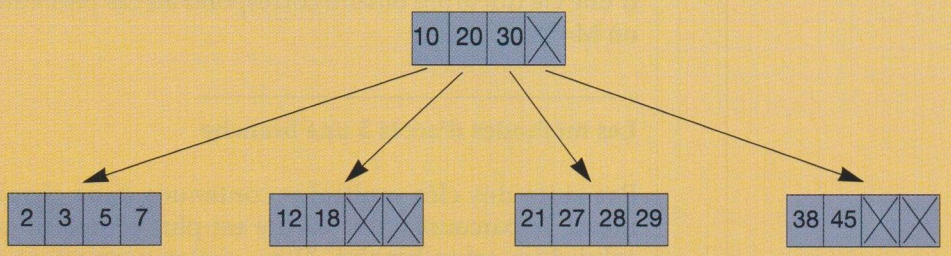


FIGURE 5 : *un B-arbre d'ordre 5*



dans un B-arbre sont conçues de manière à conserver les trois propriétés fondamentales des B-arbres, ce qui garantit notamment l'équilibre constant du B-arbre. Il arrive donc que l'insertion ou la destruction d'une clé entraîne une réorganisation partielle de la structure du B-arbre. Ces petites opérations sont toutefois préférables aux réorganisations massives imposées par d'autres méthodes d'indexation durant lesquelles l'accès à la base de données est impossible.

Insérer une clé dans un B-arbre

Comme avec les arbres binaires, l'insertion d'une clé dans un B-arbre nécessite une recherche préalable qui indique l'emplacement où devrait se trouver la clé à insérer. Si le nœud en question n'est pas plein, il suffit d'y ajouter la nouvelle valeur tout en maintenant l'ordre ascendant des clés au sein du nœud.

La figure 6 indique le résultat obtenu après l'insertion d'une clé de valeur 15 dans le B-arbre de la figure 5. Il est cependant possible que le nœud en question contienne déjà le nombre maximal de clés autorisé (soit M-1 pour un B-arbre d'ordre M). Ce cas se produit si l'on essaie d'insérer une clé de valeur 6 dans le B-arbre de la figure 6. L'élément «central» du nœud, que l'on obtiendrait après l'insertion, est alors introduit dans le nœud père du nœud plein, tandis que ce dernier se trouve divisé en deux nœuds de taille égale.

La figure 7 représente le résultat obtenu : la clé de valeur 5 est insérée dans le nœud père du nœud «saturé» <2,3,5,6,7>, et la séparation des clés restantes de ce nœud donne les nœuds <2,3> et <6,7>.

Il peut évidemment arriver que le nœud père soit plein lui aussi lorsque l'on tente d'y insérer l'élément central du nœud à diviser : le nœud père est alors divisé à son tour selon la même méthode, et cette procédure de division se poursuit autant de fois qu'il le faut. Cela peut éventuellement forcer à insérer une nouvelle clé dans la racine du B-arbre. Si cette dernière est également saturée, elle est divisée comme un nœud normal, mais la clé qui aurait dû s'insérer dans le nœud père constitue alors la nouvelle racine de l'arbre. Une division de la racine augmente donc d'un niveau la profondeur du B-arbre.

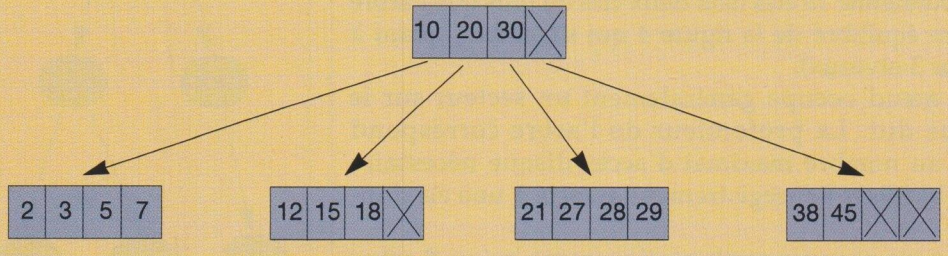
A titre d'exemple, la figure 8 représente le B-arbre qui résulte de l'insertion d'une clé de valeur 25 dans le B-arbre de la figure 7. Pour aboutir à ce résultat, le nœud <21,25,27,28,29> a d'abord été divisé en <21,25> et <28,29>, tandis que la clé de valeur 27 allait s'insérer dans la racine de l'arbre. Celle-ci se trouvant alors saturée (<5,10,20,27,30>), elle est divisée en <5,10> et <27,30>, et une nouvelle racine ne contenant que la clé de valeur 20 se trouve créée.

La destruction d'une clé dans un B-arbre

La procédure de destruction d'une clé à l'intérieur d'un B-arbre est plus complexe que l'insertion car le nombre de cas à considérer est plus important. Il faut d'abord commencer par rechercher l'emplacement de la clé que l'on veut détruire. Si celle-ci appartient à une feuille de l'arbre, il suffit de retirer la valeur dans le nœud en question.

Ainsi, le fait de détruire la clé de valeur 15 dans le

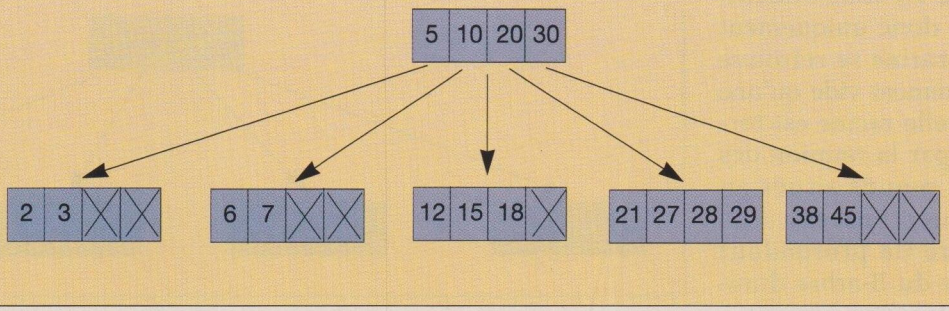
FIGURE 6 : *insertion d'une clé de valeur 15*



B-arbre de la figure 6 ramène l'arbre dans la situation de la figure 5.

Si la clé à détruire fait partie d'un nœud intérieur de l'arbre, il faut remplacer la clé détruite par une autre clé déjà présente dans l'arbre (pour conserver le même nombre de clés dans le nœud intérieur, donc un nombre identique de descendants, et par conséquent l'équilibre du B-arbre).

FIGURE 7 : *insertion d'une clé de valeur 6 et division d'un nœud*



La clé en question sera la première clé (celle qui possède la valeur la plus faible) de la feuille située le plus à gauche sur le sous-arbre de droite de la clé à détruire.

Ainsi, supprimer la clé de valeur 20 dans l'arbre de la figure 5 amène la clé de valeur 21 à sa place, comme le montre la figure 9 (la situation serait d'ailleurs similaire si l'on retirait la même clé de l'arbre représenté par la figure 8).

Il apparaît donc que la destruction d'une clé dans un B-arbre finit toujours par retirer une clé de l'une des feuilles de l'arbre. Après cette opération, la feuille en question risque toutefois de ne plus comporter suffisamment de clés pour satisfaire la deuxième propriété des B-arbres (celle qui impose un remplissage minimal de chaque nœud).

On résout ce problème en redistribuant les clés entre la feuille en sous-effectif et l'une de ses voisines, de préférence celle qui contient le plus de clés. Ce cas se produit si l'on détruit la clé de valeur 38 dans le B-arbre de la figure 9.

On réunit alors les clés qui restent dans la feuille en sous-effectif (qui contient dans notre cas uniquement la valeur 45), celles de sa voisine de gauche (qui comporte les clés <27,28,29>) et leur clé parente (de valeur 30).

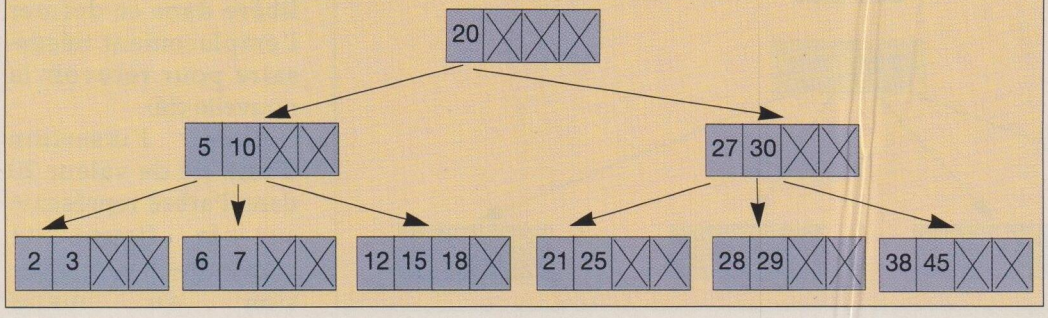
On obtient donc l'ensemble de clés <27,28,29,30,45>, que l'on redistribue de manière aussi symétrique que possible (la clé de valeur 29 devenant le parent des feuilles <27,28> et <30,45>). Le résultat de l'opération est représenté sur la figure 10.

Redistribuer les clés ou réunir les feuilles

Un autre cas particulier peut se produire si chaque feuille voisine de celle en sous-effectif contient à peine le nombre minimal de clés qui satisfait la deuxième propriété des B-arbres. Il est alors impossible de redistribuer les clés par la méthode précédente sans créer une nouvelle feuille en sous-effectif. C'est, par exemple, ce qui arrive si l'on essaie de détruire la clé de valeur 27 dans le B-arbre de la figure 10. Il faut alors réunir la feuille concernée par la destruction avec l'une de ses voisines pour former une feuille unique dans laquelle est également intégré le parent des deux feuilles réunies. Si l'on détruit la clé de valeur 27 du B-arbre de la figure 10 et que l'on réunit la feuille <12,18> avec la feuille <28> et leur parent <21>, on forme ainsi une nouvelle feuille qui contient les clés <12,18,21,28> (comme le montre la figure 11).

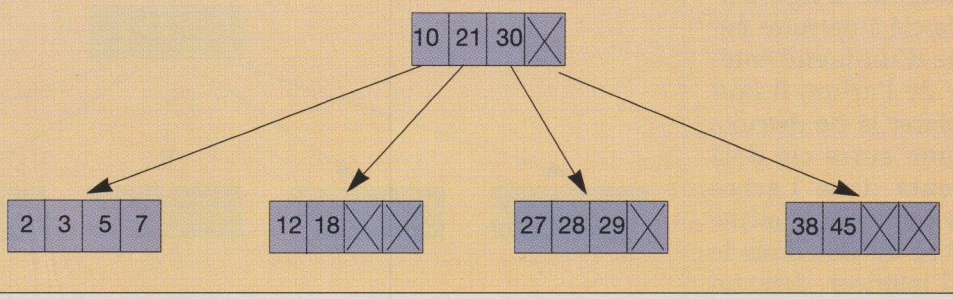
La réunion de deux feuilles oblige donc à «emprunter» une clé au nœud père des feuilles concernées. Ce nœud risque donc de se trouver en sous-effectif après cette opération, ce qui oblige alors à lui appliquer, à son tour, l'une des deux méthodes étudiées ci-dessus (redistribution des clés ou réunion de deux nœuds). Comme pour l'insertion d'une clé, ces réorganisations peuvent se propager dans le B-arbre et atteindre éventuellement sa racine. Mais, contrairement aux feuilles et aux nœuds

FIGURE 8 : *insertion d'une clé de valeur 25 et division de la racine de l'arbre*



intérieurs de l'arbre, la racine d'un B-arbre peut rester en sous-effectif. C'est donc uniquement si la racine se retrouve entièrement vide qu'une nouvelle racine est formée par la réunion des deux nœuds situés au premier niveau de l'arbre (la profondeur totale du B-arbre diminue alors d'un niveau).

FIGURE 9 : ***destruction de la clé de valeur 20 sur l'arbre de la figure 5***



Ce cas se produit si l'on détruit la clé de valeur 25 dans le B-arbre de la figure 8 : il faut réunir la feuille <21> avec la feuille <28,29> en «empruntant» la clé <27> à leur nœud père qui se trouve alors lui-même en sous-effectif puisqu'il ne contient plus que la clé <30>. Comme le seul voisin de ce nœud contient déjà le nombre minimal de clés imposé par la deuxième propriété des B-arbres, il faut à nouveau réunir ces deux nœuds en «empruntant» la clé <20> à la racine de l'arbre. Celle-ci est alors remplacée par le nœud <5,10,20,30> qui résulte de la dernière réunion opérée, ce qui ramène à la situation représentée sur la figure 7.

Variations sur les B-arbres

Il est possible de construire plusieurs types d'arbres conçus sur le même principe que les B-arbres. Ces arbres dérivés des B-arbres s'obtiennent en modifiant l'une des trois propriétés de base ainsi que les algorithmes d'insertion et de destruction de clés.

Les B*-arbres se déduisent des B-arbres en modifiant la deuxième propriété pour imposer que chaque nœud soit au moins rempli aux deux tiers de leur capacité maximale en nombre de clés (au lieu de la moitié pour les B-arbres). On obtient ainsi des nœuds mieux remplis en moyenne qu'avec les B-arbres, d'où une meilleure utilisation de l'espace disque réservé pour l'index. Cette contrainte diminue souvent la profondeur d'un B*-arbre par rapport à celle du B-arbre correspondant, ce qui améliore également la vitesse de recherche.

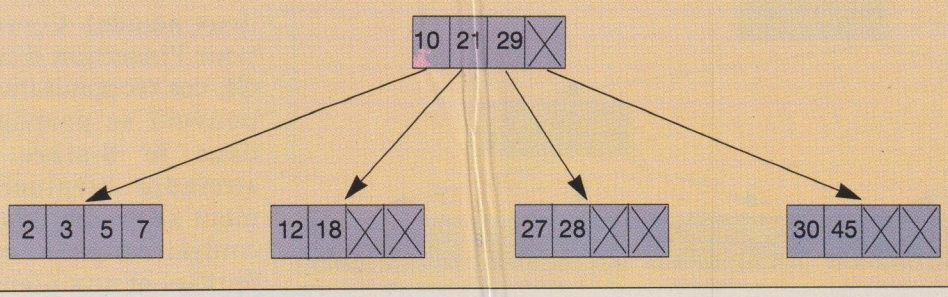
La recherche dans un B*-arbre s'effectue exactement comme dans un B-arbre, et il est possible de programmer les insertions et les destructions de clés de manière tout à fait similaire. Les insertions causent toutefois plus de divisions de nœuds dans un B*-arbre que dans un B-arbre, à cause du meilleur remplissage de chaque nœud. Il est donc avantageux de modifier l'algorithme d'insertion pour éviter de diviser un nœud si ce n'est pas indispensable, car cette opération est relativement pénalisante en termes d'accès au disque et de performances. De plus, le fait de limiter les divisions des nœuds évite parfois la propagation d'une division jusqu'à la racine de l'arbre, ce qui empêche à son tour une augmentation de la profondeur totale de ce dernier. A noter cependant que la méthode des permutations locales (local shifts), étudiée ci-après, pour limiter les divisions de nœuds peut également être appliquée aux B-arbres classiques.

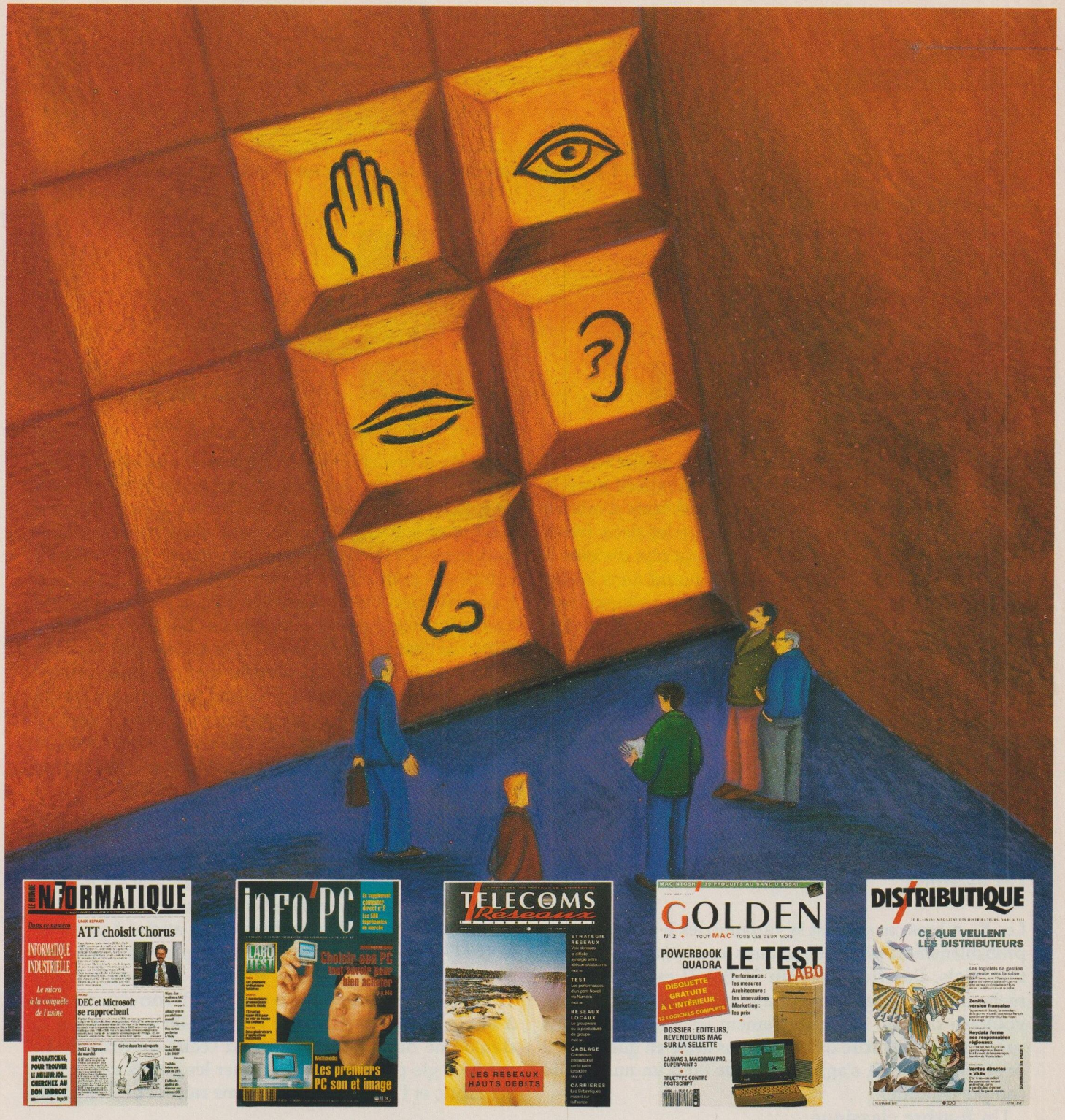
B*-arbres et permutations locales

Au lieu de diviser un nœud plein dès qu'il faut y insérer une nouvelle clé, on commence par examiner son frère de droite. Si ce nœud n'est pas plein, on lui ajoute la clé parente des deux nœuds concernés, tandis que celle-ci est remplacée par la plus grande clé du nœud plein (ce qui libère dans ce dernier l'emplacement nécessaire pour recevoir la nouvelle clé).

Ainsi, l'insertion d'une clé de valeur 20 dans l'arbre représenté sur la figure 11 n'entraîne pas la division du nœud

FIGURE 10 : ***destruction de la clé de valeur 38 et redistribution des clés***





DÉSORMAIS, L'INFORMATIQUE FAIT APPEL A VOS 5 SENS.

A chacun des publics vivant chaque jour l'aventure de l'informatique, IDG Communications apporte une réponse analysée, approfondie et objective, au travers de 5 titres ciblés et complémentaires.

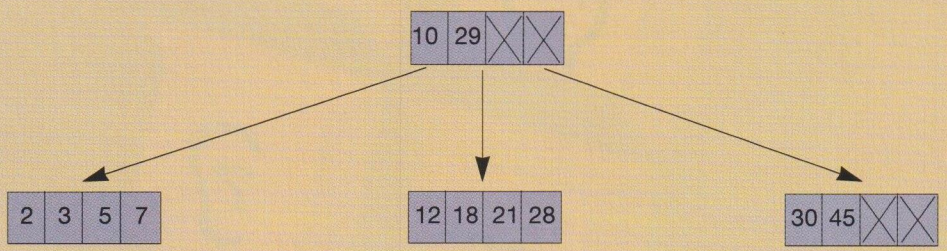
- **Le Monde Informatique pour**
  REGARDER ce qui bouge dans le Monde Informatique.
- **Info PC pour**
  TOUCHER l'innovation du bout des doigts.
- **TELECOMS & RÉSEAUX International pour**
  ÉCOUTER le souffle des nouvelles technologies.
- **GOLDEN pour**
  GOÛTER aux plus beaux fruits du Macintosh.
- **DISTRIBUTIQUE pour**
  SENTIR les tendances du marché.

IDG
COMMUNICATIONS FRANCE

DIRECT Associés

<12,18,21,28> comme cela serait le cas avec l'algorithme de base. La clé parente <29> est ajoutée au frère de droite <30,45> et remplacée par la clé <28> du nœud initialement plein. Celui-ci peut alors recevoir la clé de valeur 20, ce qui amène à la situation décrite par la figure 12.

FIGURE 11 : *destruction de la clé de valeur 27 et réunion de deux feuilles*



Si le frère de droite se trouve déjà plein (ou s'il n'existe pas), la même opération est réalisée avec le frère de gauche du nœud dans lequel on tente d'insérer la nouvelle clé. C'est seulement lorsque les deux frères du nœud considéré sont déjà remplis qu'une division a lieu. Pour cela, on forme (à partir du nœud dans lequel s'insère la nouvelle clé et de l'un de ses voisins) trois nœuds dans lesquels les clés sont réparties aussi équitablement que possible. La figure 13 montre le résultat obtenu après l'insertion d'une clé de valeur 4 dans l'arbre de la figure 12. Ces divisions peuvent se propager sur plusieurs niveaux (si le nœud père est également plein) et, éventuellement, atteindre la racine de l'arbre. Dans ce dernier cas, la division de la racine s'effectue exactement comme avec l'algorithme classique.

Les B+-arbres

Les B+-arbres constituent la variante des B-arbres qui est probablement la plus utilisée dans les systèmes d'indexation réels. En effet, avec un SGBD, il est courant de rechercher l'enregistrement associé à une clé, puis de vouloir examiner séquentiellement les enregistrements suivants (dans l'ordre déterminé par la clé en question : il peut, par exemple, s'agir d'une date ou d'un numéro de compte bancaire).

Un B-arbre classique fournit un moyen efficace pour trouver le premier enregistrement associé à une clé, mais le traitement séquentiel des enregistrements suivants, nécessite à chaque fois une nouvelle recherche à partir de la racine de l'arbre, d'où une performance médiocre.

Un B+-arbre résout ce problème en stockant, dans chaque feuille de l'arbre, deux pointeurs vers les frères de gauche et de droite de la feuille en question. Une fois positionné sur une clé quelconque dans un B+-arbre, le traitement séquentiel des enregistrements revient donc à suivre ces pointeurs à la manière d'une liste chaînée. L'organisation des données au sein d'un B+-arbre est toutefois assez différente de celle des B-arbres, car le B+-arbre contient en réalité deux types de nœuds : la racine et les nœuds intérieurs de l'arbre sont organisés comme un B-arbre (ou comme un B*-arbre) servant à guider les recherches vers les feuilles de l'arbre. Celles-ci contiennent, quant à elles, toutes les clés stockées dans l'arbre, les adresses des enregistrements associés à chaque clé et les pointeurs utilisés lors des accès séquentiels.

Cette distinction étant posée, il faut maintenant construire le B-arbre qui guide l'accès vers les feuilles. Pour cela, les N-1 valeurs que comporte un nœud intérieur ayant N enfants sont déterminées grâce à la règle suivante : «la k-ème valeur d'un nœud est égale à la valeur de la clé la plus petite que l'on atteint à partir de la (k+1)-ème branche du nœud en question (celle qui part à droite de la valeur)». Sur le B+-arbre de la figure 14, la valeur stockée dans la racine (20) correspond bien à la clé la plus petite que l'on atteint en suivant la deuxième branche de ce nœud (celle de droite). La même règle permet de déterminer les autres valeurs de cet arbre, et l'on applique le même raisonnement quelle

FIGURE 12 : *insertion d'une clé de valeur 20 et permutations locales*

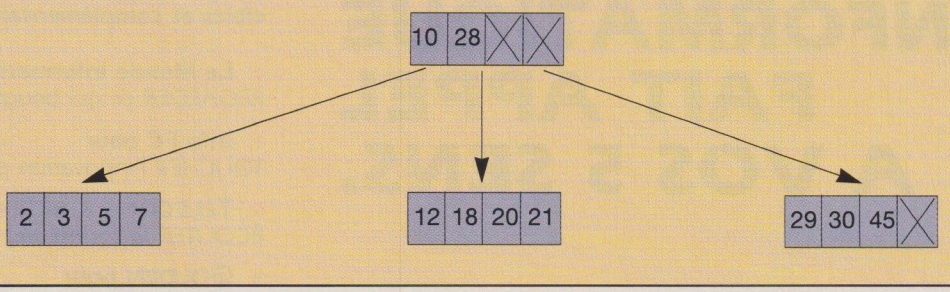
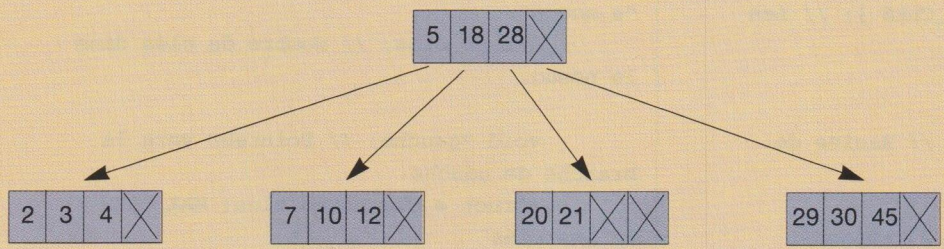


FIGURE 13 : ***insertion d'une clé de valeur 4 et division de 2 nœuds pleins en 3 nœuds***



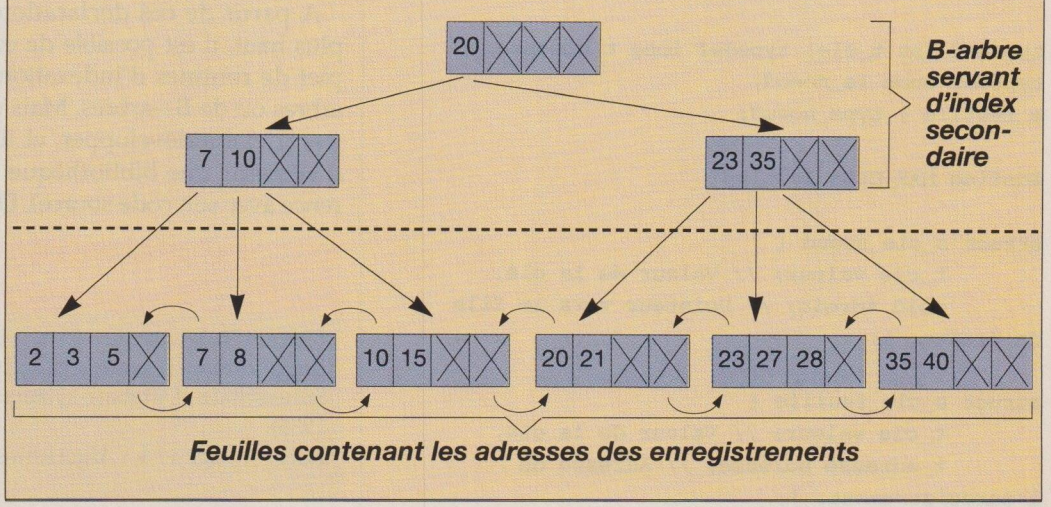
que soit la complexité de l'arbre. Une recherche dans un B+-arbre s'effectue de la même manière qu'avec les B-arbres, mais il ne faut pas s'arrêter sur un nœud intérieur dès que l'on trouve la valeur recherchée : on continue, au contraire, à progresser jusqu'aux feuilles de l'arbre, car seules ces dernières contiennent les adresses effectives des enregistrements.

La partie supérieure d'un B+-arbre (c'est-à-dire tous les nœuds qui ne sont pas des feuilles) réalise en fait une sorte d'index de deuxième niveau qui permet d'accéder rapidement à la feuille contenant les informations voulues. L'insertion et la destruction de clés dans un B+-arbre suivent les mêmes règles que pour un B-arbre ou pour un B*-arbre, en tenant compte de la modification de l'algorithme de recherche qui oblige à descendre jusqu'aux feuilles de l'arbre pour trouver l'endroit où insérer (ou détruire) une clé. Il faut néanmoins veiller à conserver la cohérence des valeurs stockées dans les nœuds intérieurs de l'arbre lorsqu'une feuille est modifiée, ce qui oblige à «remonter» les branches concernées pour ajuster certaines valeurs.

Etudions maintenant deux exemples de déclaration, en langage C, des structures de données nécessaires au stockage d'un B-arbre et d'un B+-arbre. La programmation d'algorithmes à base de B-arbres peut se faire selon des méthodes très diverses. Les exemples suivants ne fournissent qu'une illustration en langage C des concepts abordés dans cet article. Considérons d'abord le cas d'un B-arbre pour lequel les clés sont représentées par des variables de type integer et les adresses des enregistrements par des entiers longs correspondant, par exemple, au paramètre offset d'un appel à la fonction de bibliothèque lseek(). En plus de ces deux données, chaque clé contient un pointeur vers le nœud situé sur la branche de droite rattachée à cette clé. Les nœuds eux-mêmes comportent un nombre fixe de clés (déterminé par la constante MAX_CLES), un pointeur vers le nœud situé sur la branche la plus à gauche rattachée au nœud courant et un compteur qui indique le nombre de clés stockées dans le nœud. Enfin, l'arbre lui-même contient un pointeur sur le nœud racine et une zone tampon utilisée pour les réorganisations occasionnées par certaines insertions ou destructions :

```
typedef int t_cle; typedef long t_adresse;

#define MAX_CLES 10

struct s_cle {
       t_cle valeur; // Valeur de la clé.
       t_adresse adresse; // Adresse de

l'enregistrement.
       struct s_nœud *noeud_droit; //
Pointeur vers le fils de droite. };

       struct s_noeud {
       int nb_cles; // Nombre de clés dans
```

FIGURE 14 : ***un B+-arbre d'ordre 5***



```
le nœud.
        struct s_nœud *noeud_gauche;
// Pointeur vers la branche de gauche.
        struct s_cle cles[ MAX_CLES ]; // Les
clés elles-mêmes. };

typedef struct {
        struct s_nœud *racine; // Racine de
l'arbre.

        struct s_cle *buffer; // Tampon
utilisé pour les réorganisations. }
b_arbre;
```

L'exemple qui précède suppose que le B-arbre soit stocké en mémoire. Dans la pratique, les nœuds sont stockés dans un fichier d'index séparé du fichier principal (celui qui contient les données elles-mêmes). Les pointeurs de l'exemple ci-dessus sont alors remplacés par des adresses sur le disque (par exemple, les déplacements à partir du début du fichier d'index).

Pour minimiser les accès au disque et pour optimiser l'utilisation de l'espace disque, on «aligne» généralement les nœuds sur la taille des secteurs ou des clusters (groupes de secteurs que le DOS lit ou écrit en une seule opération). Le nombre de clés que l'on stocke dans chaque nœud est alors calculé en fonction de cette contrainte. Un B*-arbre utilise la même structure de données qu'un B-arbre classique, seules les routines d'insertion et de destruction sont modifiées pour tenir compte des nouvelles règles à respecter.

Le cas des B+-arbres est, quant à lui, plus complexe car les nœuds intérieurs de l'arbre ont une structure différente de celle des feuilles. Ceci oblige à se servir des pointeurs de type void * (dans le cas où l'arbre est «en mémoire») et à inclure un champ permettant de distinguer si l'on pointe sur un nœud intérieur ou sur une feuille :

```
typedef int t_cle; typedef long t_adresse;
typedef enum {e_noeud,
e_feuille } type_noeud;

#define MAX_CLES 10

struct s_cle_noeud {
        t_cle valeur; // Valeur de la clé.
        void *droit; // Pointeur vers le fils
de droite. };

struct s_cle_feuille {
        t_cle valeur; // Valeur de la clé.
        t_adresse adresse; // Adresse de
l'enregistrement. };

struct s_noeud {
        type_noeud type; // Normalement égal à
"e_noeud".
        int nb_cles; // Nombre de clés dans
le noeud.

        void *gauche; // Pointeur vers la
branche de gauche.
        struct s_cle_noeud cles[ MAX_CLES ];
// Les clés
elles-mêmes. };

struct s_feuille {
        type_noeud type; // Normalement égal à
"e_feuille".

        int nb_cles; // Nombre de clés dans
la feuille.
        void *gauche; // Pointeur vers la
feuille de gauche.
        void *droite; // Pointeur vers la
feuille de droite.
        struct s_cle_feuille cles[ MAX_CLES ];
// Les clés
elles-mêmes. };
typedef struct {
        struct s_noeud *racine; // Racine de
l'arbre.

        struct s_feuille *premiere; // Feuille
la plus à gauche.
        struct s_cle_noeud *buffer_nœud; //
Tampon utilisé pour les réorganisations.
        struct s_cle_feuille *buffer_feuille;
// Idem. }
bplus_arbre;
```

A partir de ces déclarations et des principes exposés plus haut, il est possible de construire un ensemble complet de routines d'indexation à base de B-arbres, de B*-arbres ou de B+-arbres. Mais un projet de cette taille n'est pas trivial à développer, et il sera souvent plus rentable d'acquérir une bibliothèque C du commerce (de préférence avec son code source). Débutants en C, s'abstenir. ❑

BIBLIOGRAPHIE

Sharam Hekmatpour : C++, a book and disk guide for C programmers (Prentice-Hall, 1990). H.F. Korth & A. Silberschatz : Database system concepts (Mc-Graw-Hill, 1986).
Robert Sedgewick : Algorithms (Addison-Wesley, 1983).

13 - 14 OCTOBRE 1992

COMPU SEARCH

LE SALON DE L'EMPLOI INFORMATIQUE

A LA VEILLE DE SA 9ème EDITION, COMPU SEARCH VOUS OFFRE LES CM² QUI OPTIMISERONT VOS M²

54.900F*

*** Prix d'un stand à COMPU SEARCH maintenu pour octobre 92 avec en plus, une annonce gratuite dans LE MONDE INFORMATIQUE.**

DES VISITEURS CIBLES ET MOTIVES

Entre 3000 et 5000 visiteurs sont candidats à l'emploi informatique sur chaque COMPU SEARCH.

Joints personnellement avant le salon à partir des fichiers les plus qualifiés, ils sont de formation supérieure pour plus de 70 % d'entre eux, dont 32 % d'ingénieurs.

Ils seront présents pour vous rencontrer en direct les 13 et 14 octobre prochains afin de préparer leur avenir et de contribuer par leurs compétences à votre développement.

ORGANISE PAR

IDG CARRIERES

Vos contacts :
Jean-Loup HUGODOT
Denis SEMATISKY
Tél. : (1) 49.04.79.07
Tél. : (1) 49.04.79.20

POUR EXPOSER A COMPU SEARCH/OCTOBRE 92

Bon à découper et à renvoyer à Jean-Loup HUGODOT
IDG CARRIERES – Cedex 65 – 92051 PARIS LA DEFENSE – Tél. : (1) 49.04.79.07

Société _____
Nom _____
Adresse _____
Code postal |_|_|_|_|_| Ville _____
_____
Téléphone : _____ Fax : _____

Je désire recevoir :
- ☐ Un dossier exposant.
- ☐ La visite d'un commercial.
- ☐ Une invitation en tant que visiteur.

Au fil des copies d'écran de cet article, les points critiques à ne pas négliger sont abordés. Délibérément, nous n'entrerons pas dans certains détails qui, certes, ne manquent pas d'intérêts mais qui ne sont pas essentiels. En suivant les éclaircissements fournis les uns après les autres, on est sûr de disposer d'un vade-mecum utilisable en toute circonstance.

```
Exemple :
Résultat de la somme suivante sur 16 bits :
              D8D1          (hexa)
soit  1101 1000 1101 0001   (binaire)
       100011    010001   12 bits de poids faible
       b    b    b    b
       11   6    5    0

soit      35       17   (décimal)
        + 32     + 32   Ajout de 32 (décimal) pour
          —        —    avoir un caractère affichable
          67       49   (décimal)
résultat "C"      "1"   Checksum = "C1"
                        sur 2 caractères
```

Ce traitement est effectué à l'émission, puis à la réception avec un contrôle d'égalité. Avec l'apparition des débits supérieurs à 4 800 bits/s, le contrôle checksum est devenu obsolète. La technique la plus utilisée aujourd'hui se base sur une détection d'erreur au niveau du bit. Elle est nommée contrôle de redondance cyclique (CRC: Cyclic Redundancy Checking) et utilise les polynômes mathématiques.

Prenons le cas du contrôle CRC de 16 bits sur 3 caractères, normalisé par le CCITT. Le bloc de données à transmettre est traduit sous la forme d'un polynôme. Chaque bit du bloc constitue un coefficient du polynôme de degré n-1 (où n représente le nombre de bits du bloc de données), pour donner :

$$P(x) = c_0 + c_1 x + \ldots + c_{n-1} x^{n-1}$$

Ce dernier est divisé par le polynôme générateur de degré 16 de l'appelant. Le reste de cette division, de degré maximal 15, donne :

$$R(x) = r_0 + r_1 x + \ldots + r_{15} x^{15}$$

Les 16 bits $r_0, r_1 \ldots r_{15}$ forment les 3 caractères du CRC, zone de détection ajoutée à la trame, comme ci-dessous :

```
        . 1 er car CRC . 2 em car CRC . 3 em car CRC .
DONNEES . bit à bit   . bit à bit   . bit à bit .
        . 15   12     . 11   6      . 5    0     .
```

A la réception, le même algorithme est appliqué à la trame pour reconstituer le polynôme. Ce dernier est également divisé par le même polynôme générateur. Si le reste de cette division apparaît identique à celui indiqué dans la zone CRC, la transmission s'est bien déroulée. Dans le cas contraire, la trame est retransmise.

La technique CRC filtre un grand nombre d'erreurs : celles qui portent sur un seul ou deux bits, sur des trames ayant un nombre impair de bits erronés, sur toutes les salves d'erreurs inférieures à 16 bits, et sur plus de 99,99 % des autres erreurs.

Cette technique, très fiable, a pourtant son talon d'Achille : si une erreur de transmission s'introduit dans le mot CRC, l'ensemble de la trame est considéré comme erroné.

Voici la primitive «CHK3» en langage C, utilisée par le protocole Kermit, décrit plus loin, pour calculer le CRC sur 16 bits d'une chaîne de caractères terminée par «null». Le nombre «magique» 010201 est dérivé du polynôme CRC-CCITT.

```
chk3(s) char *s; {
    unsigned int c, q;
    LONG crc = 0; /*initialisation du CRC*/
    While ((c = *s++) != '/0') '{
    if (parity) c &= 0177;
    /*Désactivation bit parité*/
    q = (crc ^ c) & 017;
    /*4 bits de poids faible*/
    crc = (crc >> 4) ^ (q * 010201);
    q = (crc ^ (c >> 4)) & 017;
    /*4 bits de poids fort*/
    crc = (crc >> 4) ^ (q * 010201);
```

protocole peut donner un résultat excellent ou médiocre, selon le type de modem employé.

Basé sur le protocole Modem7 utilisé sur le défunt système CP/M, XModem est le premier protocole à avoir été doté d'une procédure de correction d'erreurs. Il demeure le plus répandu.

Il transfère des trames de 128 octets comprenant des fanions de début et de fin, un numéro de trame et un caractère checksum. L'expéditeur se met alors en position d'attente. Du côté du destinataire, le checksum est de nouveau calculé. En cas d'égalité, un accusé de réception positif est envoyé. Dans le cas contraire, il est négatif. XModem permet de transférer des fichiers aussi bien ASCII que binaires. Facile à installer, il ne coute rien. Toutefois, ses performances restent limitées si on les compare aux nouveaux protocoles.

A noter également que XModem emploie une représentation sur 8 bits (ASCII PC), ce qui le rend incompatible avec des ordinateurs qui ne supportent que l'aphabet ASCII du CCITT n°5, sur 7 bits. Il autorise l'émission d'un seul fichier à la fois, sans aucun attribut. La taille du fichier est arrondie à la dernière trame de 128 octets. Enfin, la petite taille de ses trames et son mode de correction SW en font un outil mal adapté au transfert de fichiers volumineux.

X-Modem, les versions se suivent...

Au cours des dernières années, de nouvelles versions de XModem ont fait leur apparition : XModem CRC, qui utilise un contrôle CRC sur deux caractères au lieu du checksum, et XModem auto qui combine les deux types de contrôle. Telink, une variante de XModem, conserve les attributs des fichiers lors du transfert, et, enfin «Relaxed XModem» offre un «time out» dix fois supérieur à la temporisation d'origine.

IModem est un protocole sans détection ni correction d'erreurs. Il envoie ses trames en flux continu sans attendre d'accusés de réception, ce qui limite son emploi au transfert de petits fichiers ou bien à des fichiers plus importants via des modems avec correction d'erreurs. Une seule erreur suffit à recommencer le transfert du fichier.

Ce protocole, créé en 1981, marque une nette avancée sur XModem. YModem emploie en standard un contrôle d'erreurs CRC et autorise l'envoi de plusieurs fichiers avec leurs attributs. Il traite des trames de 1 Ko, à comparer aux trames de 128 octets de XModem, ce qui permet de réduire à un huitième le nombre de caractères de gestion de trames et d'autant les accusés de réception.

Ces optimisations font économiser plus de 60 % du temps de transfert par rapport à XModem. Pourtant, dans de mauvaises conditions de transmission, YModem constitue un mauvais choix. Il prend 8 fois plus de temps que XModem pour renvoyer une trame incorrecte. YModem est souvent confondu avec le protocole 1K XModem.

Bien qu'il utilise la même taille de trame que Ymodem, ce protocole est identique à Xmodem CRC. Il ne peut pas transmettre plusieurs fichiers consécutifs, ni même le nom les attributs de fichier. En raison de cette confusion, et pour sa particularité de transférer des fichiers par lots, le protocole YModem original a été rebaptisé YModem Batch. YModem G, une autre version de YModem, fonctionne selon le même principe de non-détection d'erreurs que IModem.

Le protocole Kermit

Kermit a été développé en 1981 à l'université de Columbia dans un but non lucratif pour promouvoir les télécommunications. Bien que gratuit, Kermit et ses filiations logicielles ne font pas partie du domaine public. Comme pour XModem, le protocole consiste à établir une session SW entre deux usagers, laquelle se réduit pour chaque fonction par un échange de trames. Chaque trame est numérotée séquentiellement, en commençant par 0 pour la trame d'initialisation. La trame d'accusé de réception hérite du même numéro que celle à acquitter.

Lorsqu'un accusé est positif, le numéro de trame suivant augmente de 1 modulo 64. Les trames sont constituées de caractères affichables, avec une représentation ASCII sur 7 bits comprise entre 20 H et 7 EH. Les caractères de contrôle, de même que les octets des fichiers binaires, sont préfixés par un caractère spécial, puis transformés en caractères affichables. Kermit se présente comme un protocole plus élaboré que XModem.

Il permet l'envoi de plusieurs fichiers par lots, avec leur attributs ainsi que leur taille exacte. Son contrôle d'erreurs se révèle plus perfectionné que celui de XModem. Mais la force de Kermit réside dans sa disponibilité sur l'ensemble des minis et des mainframes du marché. Toutefois, cette totale compatibilité ampute de beaucoup ses performances. Tout d'abord Kermit apparaît plus lent que XModem.

Cela s'explique par la plus petite taille de ses trames, de 10 à 96 octets, qui multiplient le nombre d'acquittements. Kermit encapsule par ailleurs les caractères ASCII non affichables par un autre caractère. Les caractères de contrôle, comme CTRL C, par exemple, sont précédés par le caractère # pour donner #C.

Les octets des fichiers binaires sont préfixés par le caractère &. Si les 7 bits de poids faible d'un octet coïncident avec un caractère de contrôle ASCII, ce dernier

THÉORIE

LES PROTOCOLES DE COMMUNICATION

Présentation et mise en œuvre

HAFID MAHMOUDI

En l'espace de dix ans, les outils de télécommunication ont quitté le stade de la complémentarité pour occuper un poste de premier plan. La tendance du marché va jusqu'à intégrer des fonctions de communication et de transfert de fichiers dans les applications. Une bonne connaissance des protocoles mis en œuvre par ces outils devient, par conséquent, indispensable pour ouvrir une application sur l'extérieur.

Un protocole de communication est un ensemble de règles hiérarchisées indispensables à l'utilisation d'un réseau public, comme le Réseau téléphonique commuté (RTC), ou privé, dans le cas de liaisons spécialisées.

Son rôle consiste à assurer l'intégrité d'un dialogue, lors de l'échange d'informations appelées «paquets» ou «trames». Ces règles définissent l'établissement d'une connexion, organisent la discipline des voies et, surtout, garantissent la validité des données.

Détection et correction d'erreurs

L'infrastructure du réseau téléphonique n'autorise pas une transmission sans erreurs. Selon le CCITT (Comité consultatif international des télégraphes et téléphones), pour des débits inférieurs à 1 200 bits/s, le taux d'erreur est de 5 pour 100 000 bits sur une liaison spécialisée et d'une erreur pour 1 000 bits sur le RTC. La qualité de la transmission s'avère donc insuffisante. Les protocoles permettent d'atténuer ces inconvénients. Comment se détecte une erreur?

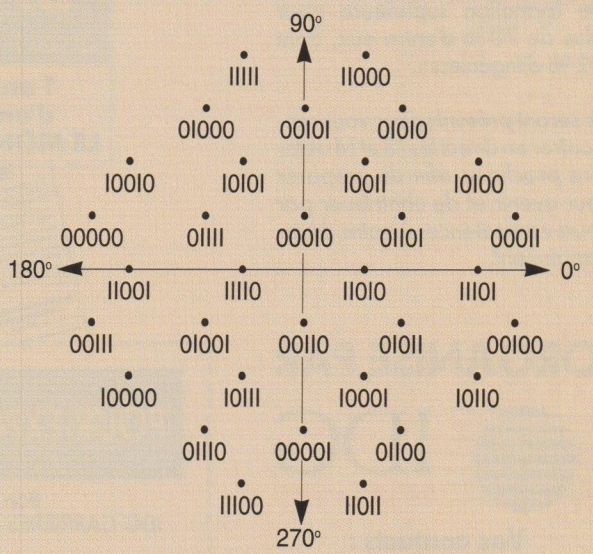
La méthode la plus simple fait appel au contrôle de parité. Elle consiste à ajouter à chaque caractère un bit de tel sorte que la somme des éléments binaires modulo 2 soit égale à 0 ou à 1. A l'arrivée, le même contrôle s'effectue, et le résultat est comparé au bit de parité de l'octet. En cas d'inégalité, l'octet est considéré comme erroné.

Ce type de contrôle présente bien des faiblesses. Par exemple, lorsque deux bits du même caractère sont erronés, le bit de parité se trouve inhibé. En outre, le bit supplémentaire pénalise le débit utile. Les premiers protocoles élaborés ont utilisé, outre le contrôle de parité, une technique de contrôle appelée «checksum» sur 1 ou 2 caractères.

Un checksum de 2 caractères correspond au 12 bits de poids faible de la somme arithmétique des valeurs ASCII de chacun des caractères. La trame est ainsi schématisée :

```
        .   1er caractère   .   2ème caractère   .
DONNEES .   b  à  b         .   b  à  b         .
        .   11    6         .   5    0          .
```

DIAGRAMME SPATIAL D'UN MODEM V32 À 9 600 BIT/S



Remarque : L'espace entre les 32 états binaires est étroit. Dans de mauvaises conditions de transmission, certains de ces états vont se confondre.
A l'arrivée, les informations seront erronées.

nication, ce qui permet d'envoyer un fichier et d'entamer un dialogue sur une autre fenêtre de travail ou encore d'ajouter d'autres fichiers au cours d'une transmission. BiModem est le protocole à utiliser avec les modems aux recommandations V32 et V32 bis.

Au pays où le Minitel est roi, il convient d'aborder le protocole de transfert propre au réseau Télétel. Ce dernier autorise une correction d'erreurs sur la voie principale à 1 200 bits/s du modem V23. Il fonctionne avec une trame de 17 octets et met en œuvre simultanément un contrôle CRC et de parité. Il est notamment utilisé dans les kits de téléchargement distribués par les serveurs de logiciels du domaine public.

Mode SW contre mode flux continu

Les protocoles les plus récents, comme HyperProtocol, ont abandonné la procédure SW au profit du transfert en flux continu ou en fenêtre d'anticipation. La raison principale est la vitesse de transfert. Une étude théorique comparative montre des écarts considérables, selon le protocole utilisé et la distance de la liaison. Les derniers protocoles sont mieux adaptés aux nouveaux modems V32 et V32 bis et aux liaisons inter urbaines.

Temps de transmission en secondes d'un fichier de 10 Ko pour quatre protocoles.

	XModem	YModem	ZModem	HyperProtocol	Comp.
TRANSMISSION					
urbaine					
2 400 bits/s	44,3	44,1	43,5	43,3	14,4
inter-urbaine					
2 400 bits/s	84,8	52,6	44,0	43,8	14,9
urbaine					
9 600 bits/s	29,7	11,9	9,5	9,4	3,1
inter-urbaine					
9 600 bits/s	70,2	20,4	10,0	9,9	3,6

Source: Dvorak's Guide to PC Telecommunications

Pourquoi ces différences de débits entre ces deux générations de protocoles? Les protocoles comme XModem, YModem ou Kermit transfèrent des trames puis attendent l'accusé de réception. Entre également en ligne de compte un nouveau paramètre, le délai de propagation d'une trame. Sur de longues distances, ce dernier peut être supérieur au temps d'émission de ladite trame. Il est en plus multiplié par deux avec l'envoi de la trame d'acquittement.

Les protocoles comme ZModem et HyperProtocol envoient un flux continu de données et attendent un acquittement unique en fin de transfert des fichiers. Ce qui les affranchit du problème précédent. Bien que YModemG apporte avec des modems auto correcteurs de type MNP les mêmes performances que ZModem et HyperProtocol, il est recommandé d'utiliser un protocole de correction d'erreurs. En effet, les modems auto correcteurs agissent en aval de la liaison. Aussi, les erreurs qui pourraient survenir sur la liaison modem-interface V24 ou sur la couche logicielle de connexion ne seront pas corrigées.

La compression de fichiers

Plusieurs utilitaires de compression de fichiers permettent de réduire le temps de transmission. Compr14, Pkarc, Pkzip, Lharc... compressent tous types de fichiers avec un rapport de un quart pour un fichier «base de données», de 1/2 pour un fichier «texte» et 2/3 pour un «exécutable». Ces utilitaires doivent être intégrés dans les procédures de transmission avant et après le transfert. Avec l'apparition des modems MNP, ce travail de compression/décompression devient transparent pour l'utilisateur.

Revers de cette technique : les algorithmes micro codés de compression utilisés sont plus simples que ceux implantés dans les logiciels. Le taux moyen de compression atteint par conséquent un maximum de 50%. La compression ne s'effectue que si les deux modems sont de type MNP. De plus, l'option MNP n'est pas gratuite, à l'inverse des logiciels de compression que l'on trouve dans le «domaine public». Certains protocoles possèdent leur propre méthode de compression.

C'est le cas de Kermit et de JModem qui emploient un algorithme qui agit sur la répétition d'un même caractère. Celui-ci remplace chaque caractère qui apparaît consécutivement plus de trois fois par une séquence de trois caractères, un caractère indicateur, un compteur et le caractère répété. Cette méthode autorise un taux de compression de 10% à 20%. HyperProtocol intègre le même algorithme que les logiciels spécialisés de compression. Il s'agit de la méthode Ziv-Lempel-Welch.

Le contrôle de flux

Les modems auto correcteurs stockent temporairement les données dans un tampon (buffer) pour correction. Cela implique que le terminal transmette les informations avec un débit supérieur à celui du modem, avec pour conséquence aléatoire un engorgement du buffer. La gestion de ce problème s'effectue selon deux méthodes. La première, de nature logicielle, se sert de deux caractères de contrôle appelés «XON» et «XOFF». Lorsque le buffer est plein, le modem envoie au termi-

est précédé de &#. L'effet s'avère désastreux, chacun de ces caractères étant multiplié par deux ou par trois.

Quant aux acquittements, ils se composent d'une trame entière au lieu d'un seul caractère pour XModem. Précisons que Kermit met en œuvre un algorithme de compression simple qui atténue légèrement ces faiblesses. En résumé, le débit utile de Kermit est égal à environ la moitié, voire un tiers de celui de XModem. De nouvelles versions de Kermit autorisent des tailles de trame pouvant aller jusqu'à 9 Ko, et la version de Kermit avec fenêtre d'anticipation permet d'éviter les temps morts du mode SW.

En 1986 apparaît un protocole d'envoi en flux continu avec correction d'erreurs, ZModem. Il insère des codes de contrôle par intervalles et marque une pause pour attendre un accusé de réception uniquement en fin de fichier. Seules les trames erronées sont exigées à nouveau. Lors d'erreurs répétées, la taille des trames est réduite à 64 octets : il s'agit du mode échelonné. En outre, ZModem a été le premier protocole à introduire une procédure de reprise en cas de coupure de la liaison. Le transfert reprend exactement à l'endroit où il a cessé. Cette technique d'envoi en continu dépasse de loin le mode SW sans contreparties négatives. Elle garantit une efficacité proche de 98%.

Les protocoles propriétaires

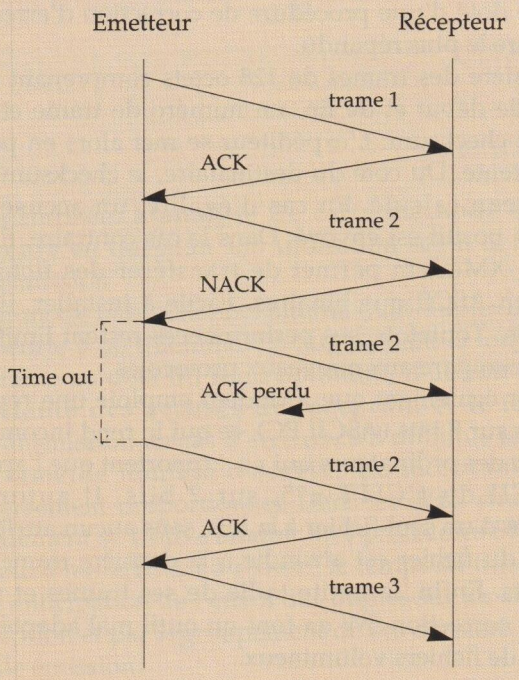
Blast a été le premier protocole de transfert bidirectionnel. Il a été développé par la société Communication Research Group en 1983. Il s'agit d'un protocole de type SW présent sur un large éventail d'ordinateurs. Créé en 1987 par la société Hilgreve, HyperProtocol a littéralement bouleversé le petit monde des protocoles de télécommunications sur PC.

Il est le premier à avoir intégré une compression en temps réel des données transmises, avec un coefficient multiplicateur pouvant aller jusqu'à 4, ce qui porte le débit théorique des modems V32 à 40 000 bits/s. La compression utilise l'algorithme «Ziv-Lempel-Welch» qui fonctionne sur tout type de fichier. Elle devient automatiquement inopérante lorsque le fichier se trouve déjà compressé. Même sans opération de compression, HyperProtocol reste intéressant.

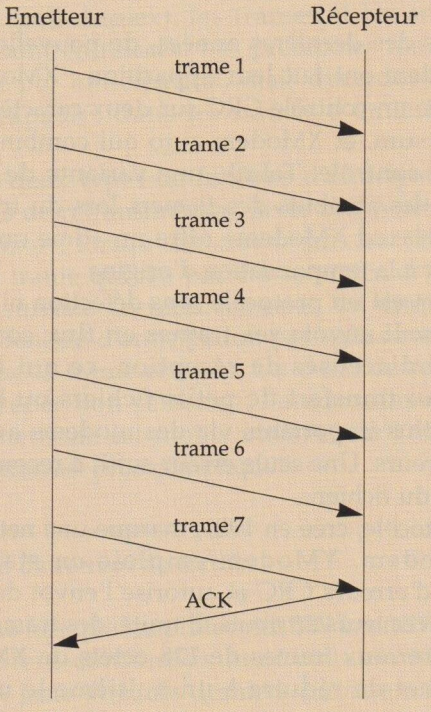
Il possède les mêmes atouts que ZModem, avec l'avantage de n'attendre qu'un seul acquittement pour plusieurs fichiers. En conclusion, HyperProtocol est à employer en priorité avec des modems rapides.

Créé en 1988, JModem constitue un protocole simple avec une représentation sur 8 bits et une procédure de compression de type Kermit. BiModem, contemporain de JModem, est plus sophistiqué en étant un protocole bidirectionnel. Il possède deux procédures de commu-

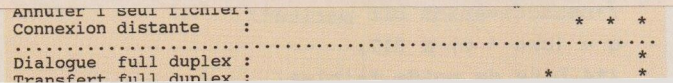
PROCÉDURE SW ET AVEC FENETRE D'ANTICIPATION



Procédure SW (Envoyer et Attendre)

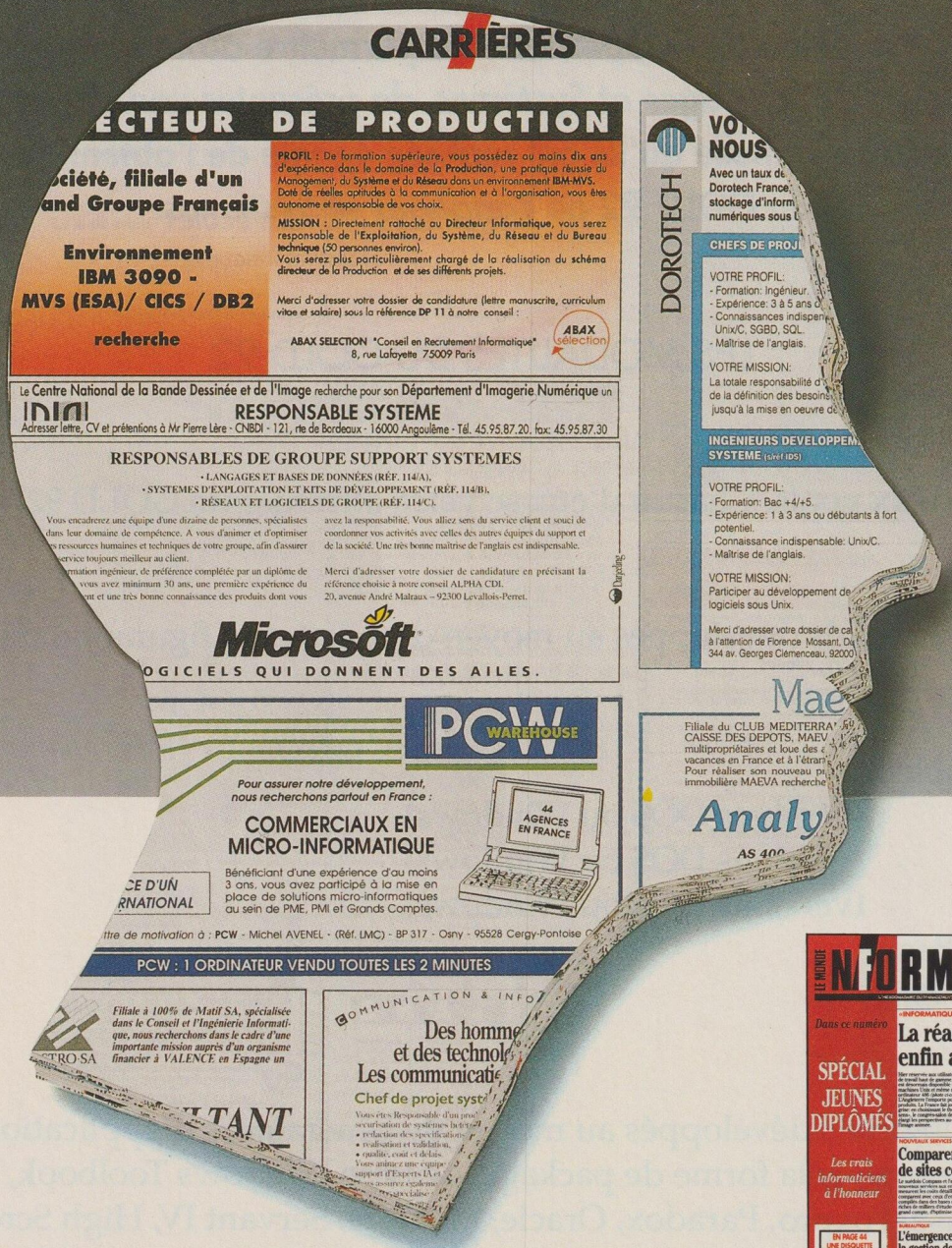


Procédure avec fenêtre d'anticipation et groupement des acquittements.



trames si une erreur vient à «couper» le flot des données. Le «mode échelonné» consiste à réduire la taille des trames en cas d'erreur, jusqu'à ce que la liaison

PROFIL EXIGÉ POUR FAIRE FACE A VOTRE CARRIÈRE



LES ENTREPRISES PERFORMANTES SONT CHAQUE SEMAINE DANS LE MONDE INFORMATIQUE.

LES PLUS BELLES CARRIERES AUSSI.

Les Annonces Classées
LE MONDE INFORMATIQUE
L'HEBDOMADAIRE DU MANAGEMENT DES SYSTÈMES D'INFORMATION

nal un XOFF pour lui demander d'arrêter le transfert de données. Le buffer dispose alors du temps pour se vider et transmet un XON au terminal pour qu'il reprenne le traitement.

Inconvénient principal, cette méthode ralentit le débit et oblige l'appelé à filtrer ces caractères. Pour éviter ces contraintes, un second mode de gestion est possible au niveau physique. Deux circuits, 105 et 106, présents sur la prise V24, jouent le même rôle que XON/OFF. Ils sont appelés respectivement RTS (Request to Send) et CTS (Clear to Send). Lorsque le terminal est prêt à recevoir, il active RTS. Quand le modem est en état de recevoir, il active le CTS. Inversement, le terminal contrôle le circuit CTS, avant d'émettre, et le modem contrôle le RTS.

Communiquer sans risque

ZModem est le premier protocole à avoir intégré une procédure de reprise en cas de coupure de la liaison. Cette possibilité permet de redémarrer le transfert de fichiers au point de rupture. Côté destinataire, le protocole contrôle l'existence du fichier.

Dans l'affirmative, et sous réserve que le fichier trouvé possède les mêmes date et heure de création, avec une taille inférieure, le protocole distant en informe l'expéditeur. Celui-ci reprend la transmission là où elle a été interrompue. BiModem va plus loin avec sa méthode de vérification de reprise. Il renvoie les CRC afin que le récepteur détermine les parties du fichier qui diffèrent. L'appelant n'a plus qu'à renvoyer les trames incorrectes.

Depuis quatre ans des modems V32 à 9 600 bits/s sont commercialisés sur le marché français. Les V32 bits/s à 14 400 bps ont vu le jour par la suite avec des fonctions «hard» d'auto corrections et de compressions. Elles ont été proposées tout d'abord en option, pour des raisons évidentes de coût et sont désormais souvent présentes en standard dans l'offre des constructeurs. La fonction corrections qu'elle soit «hard» ou «soft» est indispensable à l'utilisation de modems à 9 600 bits/s.

En effet, le réseau téléphonique commuté, avec son canal bridé sur une bande passante de 300 à 3 400 Hz, digère difficilement des débits supérieurs à 2 400 bits/s. Au-delà, il devient nécessaire d'employer des lois de codage à base de polynômes générateurs qui multiplient le nombre d'états binaires par une modulation combinée de la phase et de l'amplitude du signal électrique.

Conséquence désagréable de cette technique de modulation, lors d'une mauvaise liaison téléphonique, le modem récepteur a beaucoup de mal à reconstituer le signal de données d'origine. Ceci rend la détection et la correction d'erreurs indispensables. Les protocoles «soft» peuvent continuer à être utilisés simultanément aux fonctions internes des modems. Dans certains cas, il convient d'ajuster le time out du logiciel de communication, voire du modem, pour attendre les acquittements du destinataire. Quant à la compression «hard», sa raison d'être est beaucoup plus marketing. Elle permet de gonfler les débits indiqués sur les prospectus commerciaux et sera avantageusement remplacée par un traitement «soft».

Quant au protocole MNP (Microcom Networking Protocol), il offre plusieurs niveaux de fonctions. Les classes MNP 1, 2 et 3 assurent une détection d'erreurs simple. Les classes supérieures augmentent le débit efficace par un contrôle plus sévère avec MNP 4 et par compression avec MNP 5. Ces protocoles sont aujourd'hui les plus courants sur le marché des modems. Dernier-né, le V42 normalisé par le CCITT combine une technique propre de correction, le LAP M (Link Access Procedure M) et le protocole MNP 4 de Microcom afin d'assurer une compatibilité avec ce standard. Toutefois, les futures filiations de V42 se baseront uniquement sur le LAP M.

La programmation des protocoles

Par souci de portabilité, les protocoles de communication ont été écrits en langage C. Les programmes se trouvent segmentés en modules : composition des trames, gestion des entrées-sorties (Port et Modem), contrôle d'erreurs Checksum et CRC, gestion du terminal, gestion des fichiers...

On citera principalement les programmes sources originaux de ZModem : ZM.C pour les primitives du protocole, RBSB.C et CU.C pour la gestion du terminal respectivement sous Unix et MS-DOS, SZ.C (Send ZModem) pour le programme d'émission et RZ.C (Receive ZModem) pour le programme de réception. Ces programmes sont également compatibles avec les protocoles antérieurs X/YModem.

Les programmes sources originaux de Kermit se composent de : LCKMAIN.C pour le programme principal d'initialisation des variables, LCKPROT.C, LCKFNS1.C, LCKFNS2.C, LCKFNS3.C pour les primitives du protocole, LCKUSER.C pour les fonctions du terminal, LCKFIO.C pour la gestion des fichiers... Ils sont maintenus et peuvent être obtenus soit gratuitement, auprès des créateurs (Chuck Forsberg Omen Technology Inc) pour ZModem, Frank da Cruz, Columbia University Center for Computing Activities pour Kermit), soit téléchargés à partir de serveur comme CompuServe, CalvaCom... ou encore achetés auprès d'associations d'utilisateurs «freeware/shareware» comme OUF. ❐

OPEN DÉVELOPPEMENT

RÈGLEMENT DU GOLD

Date limite de dépôt
Envoi de votre dossier d'inscription accompagné de votre logiciel à Langages et Systèmes.
Date limite : 31 août 1992.

Dossier
Votre dossier doit comporter les pièces suivantes :
1- La fiche d'inscription
Dans le cadre d'une participation en équipe, l'inscription doit se faire au nom du responsable du projet. Une fiche d'inscription sera remplie pour chaque logiciel présenté.
2- La fiche d'identité du logiciel
Une fiche d'identité doit être remplie pour tout logiciel présenté.
3- Deux exemplaires de votre logiciel sur disquette (de préférence 3,25 pouces)
4- Les sources du programme sur disquette
5- Une version de démonstration de votre logiciel
Celle-ci sera présentée sur notre serveur 36 16 INFOPC.

Pour obtenir votre dossier d'inscription, adressez votre demande à :
GOLD / Langages et Systèmes
Cedex 65-92051 Paris la Défense

Les domaines d'application concernés
Les applications développées pourront concerner les domaines suivants : gestion, scientifique, enseignement assisté, bureautique et utilitaires systèmes.

Les critères de sélection
- Intérêt pour les utilisateurs - Originalité de l'application - Convivialité de l'application - Documentation et aide en ligne - Ergonomie de l'interface - Intégrité de l'application - Exploitation des ressources matérielles.

Le jury
Le jury du GOLD sera constitué des personnes suivantes :
- Le rédacteur en chef de Langages et Systèmes
- Notre partenaire éditeur Microsoft
- Un représentant de chacun de nos partenaires du concours : Victor, Commodore et Borland.
Les décisions prises par le jury seront sans appel.

Les prix
Dans chacun des pools définis, le vainqueur se verra décerné le GOLD 92 de Langages et Systèmes.
1er prix : Victor Technologies offrira une station multimédia pour le vainqueur du premier pool. Commodore offrira un PC 486 au vainqueur du second pool. PC Soft offrira des packages de développement DOS et Windows. Microsoft et Borland offriront des packages de développement.
2e et 3e prix : Microsoft et Borland offriront des packages de développement. PC Soft offrira High Screen DOS et Windows.
La remise des prix se fera au cours du salon Software Development qui aura lieu du 15 au 17 octobre 1992.

Langages et Systèmes *se réserve tout droit d'annulation du Grand Open Logiciel et Développement et dégage toute responsabilité pour la non-remise des prix mentionnés ci-dessus.*

info'PC

LE MAGAZINE DE LA MICRO-INFORMATIQUE PROFESSIONNELLE
Langages et Systèmes
Immeuble La Fayette 2, place des Vosges, cedex 65, 92051 Paris-la Défense 5
Tél : 49 04 79 00 - Télex : 613 234 F MONDINF

Directeur : Jacques TILLIER assisté de Catherine BOISADAM
Rédacteur en chef : François Yves LE GAL, **Assistant de la rédaction :** Serge DEL'HOMMEAU
1re secrétaire de rédaction : Catherine OLZCHANECH
Secrétaires de rédaction : Elisabeth DEROFF-VASSEUR, Isabelle DUMONTEIL, Stéphanie GRIN
Assistant de la fabrication : Stéphane GAUTIER
Ce numéro a été réalisé sous la responsabilité de Yves Guittard
Ont également collaboré à ce numéro
Thierry Bienfait (rédacteur-réviseur), Gilles Chauvaux, Maxime Daddler, Affid Mahmoudi, Eric de Kervalec, Fabian Pallini, Charles Petzold (traduction du *MSJ*), Henry-Luc Planchat (traducteur), Arnaud de la Pommeraye, Jean-Paul Pruniaux, Richard Hale Shaw (traduction de *Borland Language)*, Tom Swan (traduction de *Borland Language).*

La rédaction d'INFOPC dégage sa responsabilité en ce qui concerne les documents écrits, les matériels et les logiciels qui lui seraient envoyés sans demande préalable de sa part.

PUBLICITÉ : Directeur de la publicité : Sophie TALAYRACH, **Directeur adjoint de la publicité :** Nathalie TESSLER
Chefs de publicité : Virginie BRIL, Thien GILBERT, Hervé LABATTUT, **Assistantes de publicité :** Aromi RAJON, Laurence CARTON
DIFFUSION : Directeur : Jean PAITEL; **Responsable service abonnements :** Lucienne BOSSER
assistée de Josianne LIBAN, Rose MOLNAR, Alain PACCAGNELLA, Bernadette THUAL; **Responsable des ventes :** Pierre-Olivier PROVENT
DOCUMENTATION : Nathalie Benhaim, Jennyfer Banach, **Minitel :** 36 28 00 75

Abonnements : 49 04 79 00, 49 04 79 00; Suisse : MADITEC SA; Case postale-144CH 1020 RENENS VD1-; Tél : (021) 49 04 79 00

InfoPC est une publication éditée parMicropresse S.A. au capital de 250 000 F, en location-gérance auprès de

IDG COMMUNICATIONS FRANCE

ISSN 0981 - 6402
Commission paritaire n° 67338

INDEX PUBLICITAIRE

AD LINK	31
ALL PHASE	87
ALTISOFT	50
A.R.N.	63
CAP SESA	95
C.X.P.	33
FRANCE TEASER	26
INNOSOFT	37
LOGIDATA	4
L'A.I.I.	98
MICRO SIGMA	12,99
MICROSOFT	3,15
NEXT COMPUTER	6
NOVELL	23
N.S.L.	57
O2 TECHNOLOGY	11
OSER	54
P7I DEVELOPPEMENT	10
PARALLAX SOFTWARE	124
PC SOFT	5
PC TECHNOLOGIE	51,81
SMARTLINE	8
SOFTWARE DEVELOPPEMENT	123
SYMANTEC	43,45
S.P.C.	13

Clipper, dBase, Foxpro, Paradox, Oracle, Magic II, Servant IV, High Screen, etc. Dans ce pool, l'utilisation d'un langage interfacé ne devra pas excéder 20% du

EXPOSITION - CONFERENCES DEDIEES AUX PROGICIELS DE DEVELOPPEMENT

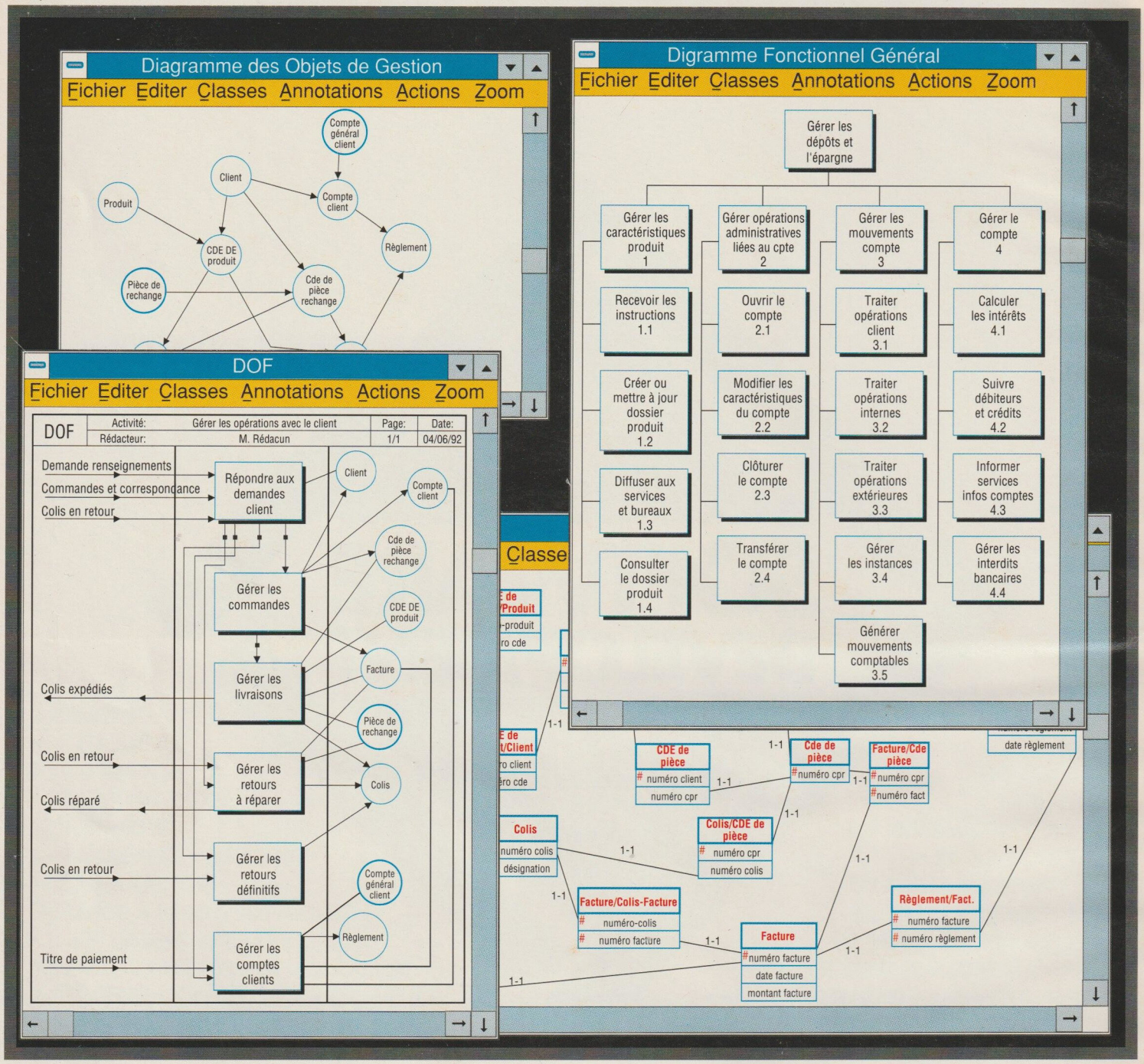
A VOIR AU SALON :
PLANIFICATION STRATEGIQUE • SYSTEME D'INFORMATION • GESTION DE PROJET • DOCUMENTEUR • ATELIER DE GENIE LOGICIEL • STRUCTURE D'ACCUEIL • DICTIONNAIRE DE DONNEES • EDITEUR DE TEXTE SOURCE • EDITEUR MULTIMEDIA • EDITEUR DE BIT-MAP • L 3G • L 4G • LANGAGE OBJET • COMPILATEUR INTERPRETEUR • INFOCENTRE • MAQUETTAGE / PROTOTYPAGE • GENERATEUR D'INTERFACE DE PRESENTATION GRAPHIQUE • MOTEUR D'INFERENCE • GENERATEUR DE SYSTEMES EXPERTS • GENERATEUR D'APPLICATION • GENERATEUR DE DOCUMENTATION • GENERATEUR DE DIALOGUE • GENERATEUR DE CONFIGURATION • GENERATEUR D'OBJET • CONTROLE QUALITE • OUTIL DE MISE AU POINT • TEST DE NON REGRESSION • ANALYSEUR DE QUALITE • SGBD / SGBDR • EXTRACTEUR DE DONNEES • UTILITAIRES DE TRI • RESTRUCTURATION DE CODE • REVERSE ENGINERING • RESEAU LOCAL • STATION DE TRAVAIL • CONSEIL, FORMATION, METHODOLOGIE • UTILITAIRES LIES AUX SGBD • OUTILS MATHEMATIQUES •

CAPRiC
GROUPE BLENHEIM
22 RUE DU PRESIDENT PRESIDENT WILSON
92532 LEVALLOIS-PERRET CEDEX
TEL : (33 - 1) 47 56 50 20
FAX : 47 56 11 90

CXP
la compétence en progiciel
19 /21 RUE DU ROCHER - 75008 PARIS
TEL : 43 87 90 28
FAX : 44 70 91 10

Service lecteurs n° 2 (dernière page)
RCS Paris B 344 500 616

GraphTalk Axial



Modèles supportés: DFG, DOG, SR, DOF, DTU, DOT,

Documentation automatique personalisable par modèle, sur l'ensemble des modèles,

Génération du shéma relationnel à partir du DOG et génération des DDL pour les principaux SGBD du marché, à partir du shéma relationnel,

Plateformes: PC, PS, stations de travail supportant OS2/PM, UNIX/MOTIF, MS/Windows,

Autres ateliers disponibles: MERISE, AXIAL, NIAM, SA, SSADM, IDEF, SD, ROOT, OOA, OOD, OSSAD.

VERSION DÉVALUATION SUR DEMANDE.

5, rue Bellini
92806 Puteaux Cedex France
Tél.: (1) 46 96 07 07 Fax: (1) 46 96 07 10

Parallax
Software technologies

SL Communication (1) 60 80 94 00 Service lecteurs n° 3 (dernière page)